NAT'L INST. OF STAND & TECH R.I.C.

A11104 256946

NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY
Research Information Center
Gaithersburg, MD 20899

NISTIR 88-4004

Product Data Exchange Specification
First Working Draft

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
National Engineering Laboratory
Center for Manufacturing Engineering
Factory Automation Systems Division
Gaithersburg, MD 20899

December 1988

National Bureau of Standards became the
National Institute of Standards and Technology
on August 23, 1988, when the Omnibus Trade and
Competitiveness Act was signed. NIST retains
all NBS functions. Its new programs will encourage
improved use of technology by U.S. industry.

Chair, IGES/PDES Organization
Bradford Smith

Coordinator, IGES/PDES Organization
Gaylen Rinaudot

U.S. DEPARTMENT OF COMMERCE
C. William Verity, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Ernest Ambler, Director

1. INTRODUCTION

1.1 TEAM HISTORY

The Cal Poly Task Team Extension is a continuation of a previous modeling effort which had been conducted at California Polytechnic University in Pomona, CA from January to April, 1987. The Extension Team was organized and chartered by George Goldsmith, John Zimmerman, and Curtis Parks in May 1987. The intent was to develop a data model that would address the issues associated with the physical description of electrical products. This work was a logical continuation of the work which was conducted at the January to April workshops wherein a data model addressing the functional behavior of electrical products had been produced.

Organizationally, the IEEE parentage of the Extension Team became somewhat tenuous after the breakup of the Design Automation Standards Subcommittee (IEEE/DASS) electrical committee in June. However, the model found a new home under the direction of Steven Piatz, head of the DASS Working Group on Information Modeling.

Within the Product Data Exchange Specification (PDES) organization, the model has been submitted to the Electrical Applications Subcomittee (EAC), chaired by Larry O'Connell. Both models are expected to become integrated into the PDES Logical Layer model.

1.1 TEAM MEMBERSHIP

The following formal roles were assigned to members of the Task Team Extension. Lead Modeler, John Zimmerman; Team Adminstrator, George Goldsmith; Technical Consultant, Roger Gale. Informally, team members fell into specific roles: Dzung Ha, database development; Kaz Hori, Query development; Joan Tyler, CIM specialist and JANUS data administrator; Robert Meagher, technical discussion and hybrid development; Mary Kathryn Kennedy, technical discussion and adminstrative support; Noel Christensen, technical consultant for topology; and Scott Madsen, technical discussion and graphic support. Curtis Parks served as Design Specialist and Editor-in-chief of the final report. Assistance was provided by Larry O'Connell of Sandia National Labortories, and the following people provided technical support at one or more sessions: Steven Piatz, Yosef Haridim, Bill Loye, Paul Nelson and Lt. Eric Gunther of the USAF CIM Office.

1.2 ORGANIZATION OF THIS REPORT

The body of this report contains this introduction, a summary, the conceptual data model developed by the task team, conclusions and recommendations. The bulk of the report body is the conceptual data model which was developed by the Extension Team. The model was partitioned into several views which reflect the logical organization of the model. A brief narrative has been included on each view of the model which articulates the important concepts contained in each view. Some 'For Exposition Only' (FEO) subsections of the model and miscelleanous illustrations have been included when these have been found to be explanatory. The distinction between a view and a FEO is that a view of the model is an exact subset of the model, whereas a FEO may contain a subjective represention of the model. That is, a FEO need not adhere to a portrayal of a subset of entities, attributes, and relationships included in the formal model nor to the model's various formalities. Other documentation of the model includes definitions for the entity pool which can be found in the Model Glossary and a set of business rules which describe the relationships among entities and their cardinality.

2. SUMMARY

2.1 PURPOSE OF THE CAL POLY TASK TEAM EXTENSION

The intention of the the Extension Team is to produce a data model which will serve as a starting point for developing industry-wide computer integrated manufacturing (CIM) environments for electrical products. Our model is based on the assumption found also in the PDES Planning Model that there is a logical separation of the functional from the physical description of electrical products. The previous modeling group which met at California Polytechnic University (Cal Poly) from January to April, 1987, produced a data model that described the functional behavior of electrical products. (This would be the data conveyed by a schematic or boolean construct and net list.) The work of the Cal Poly Task Team Extension continued this work by describing the physical nature of layered electrical products such as printed circuit boards, hybrid microcircuits, and thick film assemblies. Figure 1 uses the IDEF1X modeling methodology to describe the relationship of functional and physical data, together with requirements. This view was defined early in the original Team's work. Figure 2 illustrates the CIM environment that the model is intended for. Figure 3 describes the range of electrical products over which the Cal Poly Task Team Extension (CPTTX) model is applicable.

2.2 SCOPE OF THE CAL POLY TASK TEAM EXTENSION

The Cal Poly Task Team Extension started with an initial scope to include *all* the data that is delivered to manufacturing from design and from planning. The purpose of the model has been illustrated using two different scoping tools which are both presented here. Figure 4 portrays the scope of the Cal Poly Task Team Extension as it is expressed over a period of time. Figures 5 and 6 portray the scope of the Cal Poly Task Team Extension as seen from a process-driven or activity-oriented approach.

The diagrams found in Figures 5 and 6 are IDEF0 Activity diagrams which describe the scope of the modeling effort by describing the set of data that is available at particular points in the activity. Notice that there is no time line associated with this activity-driven approach, however, there is an implicit need to consider that these activities occur over a period of time (which may affect the data somewhat).

Appendix A1 contains a copy of the Team's charter which further defines the goals of the Cal Poly Task Team Extension.

2.3 MODEL REVIEWS

An informal review and interim workshop was hosted by Gene Lish in November 1987, at the US Naval Weapons Electronics Manufacturing Productivity Facility (EMPF) in China Lake, CA. At this review, data requirements for the automation of electrical product assemblies were discussed.

A full scale review of the model was conducted during the convening of the Electrical Applications technical subcommittee at the IGES/PDES quarterly meeting in January 1988. At this review three significant issues were raised which have been appended to the Issue List (Appendix A4). The last of these three issues was not discussed at the January review and has been included in this final report as an unresolved change to the model. (See issue issue # 16.)

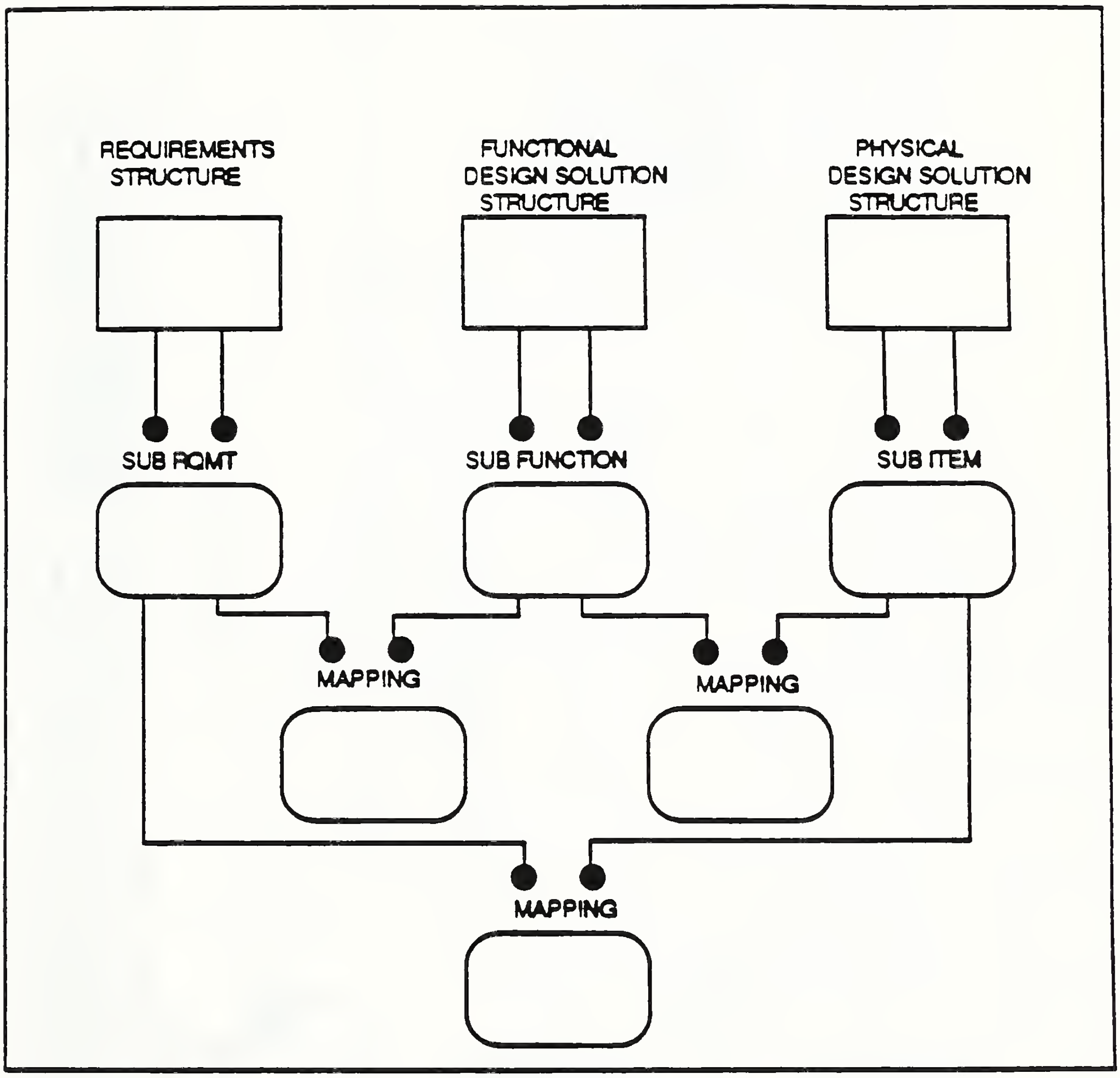
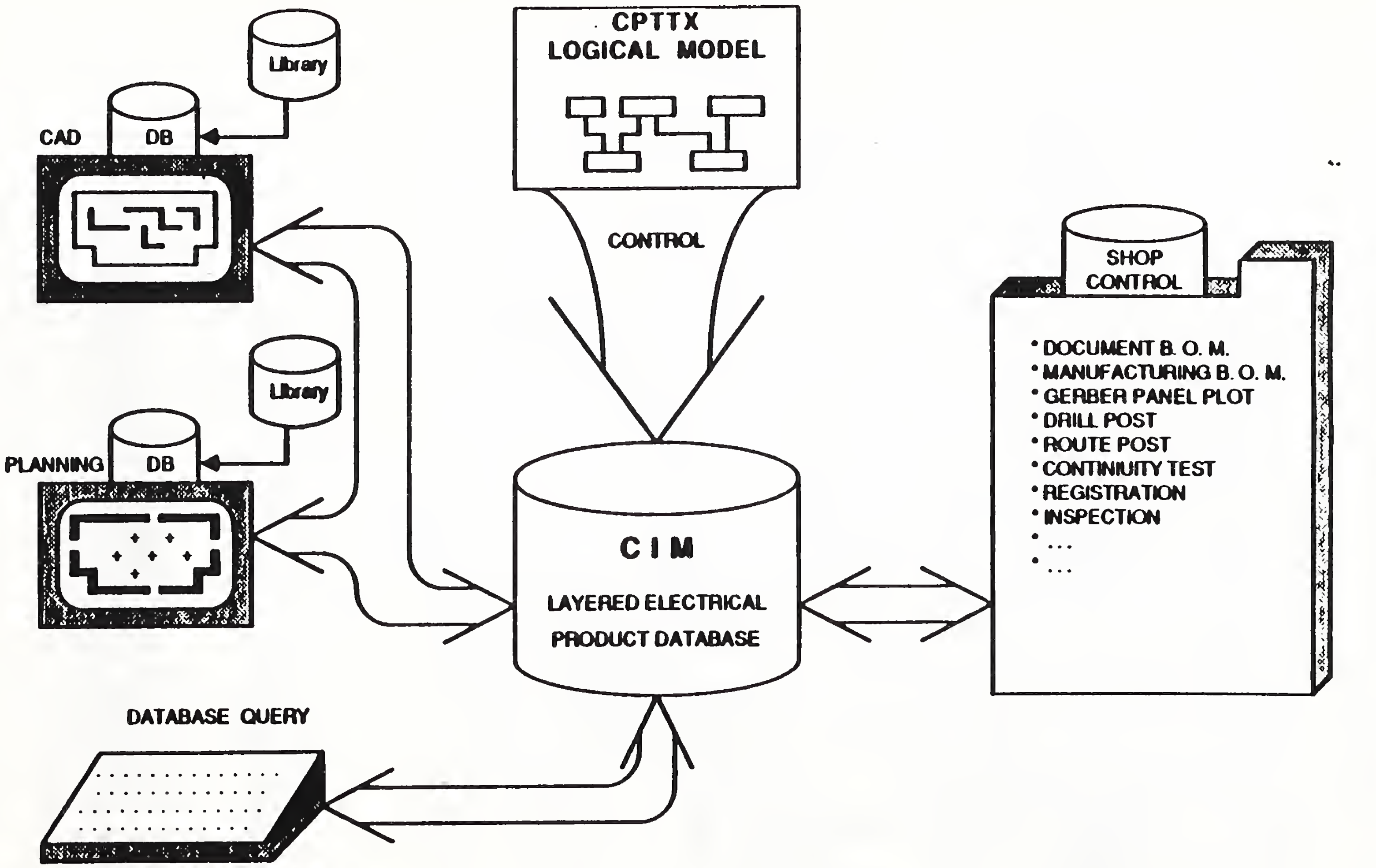


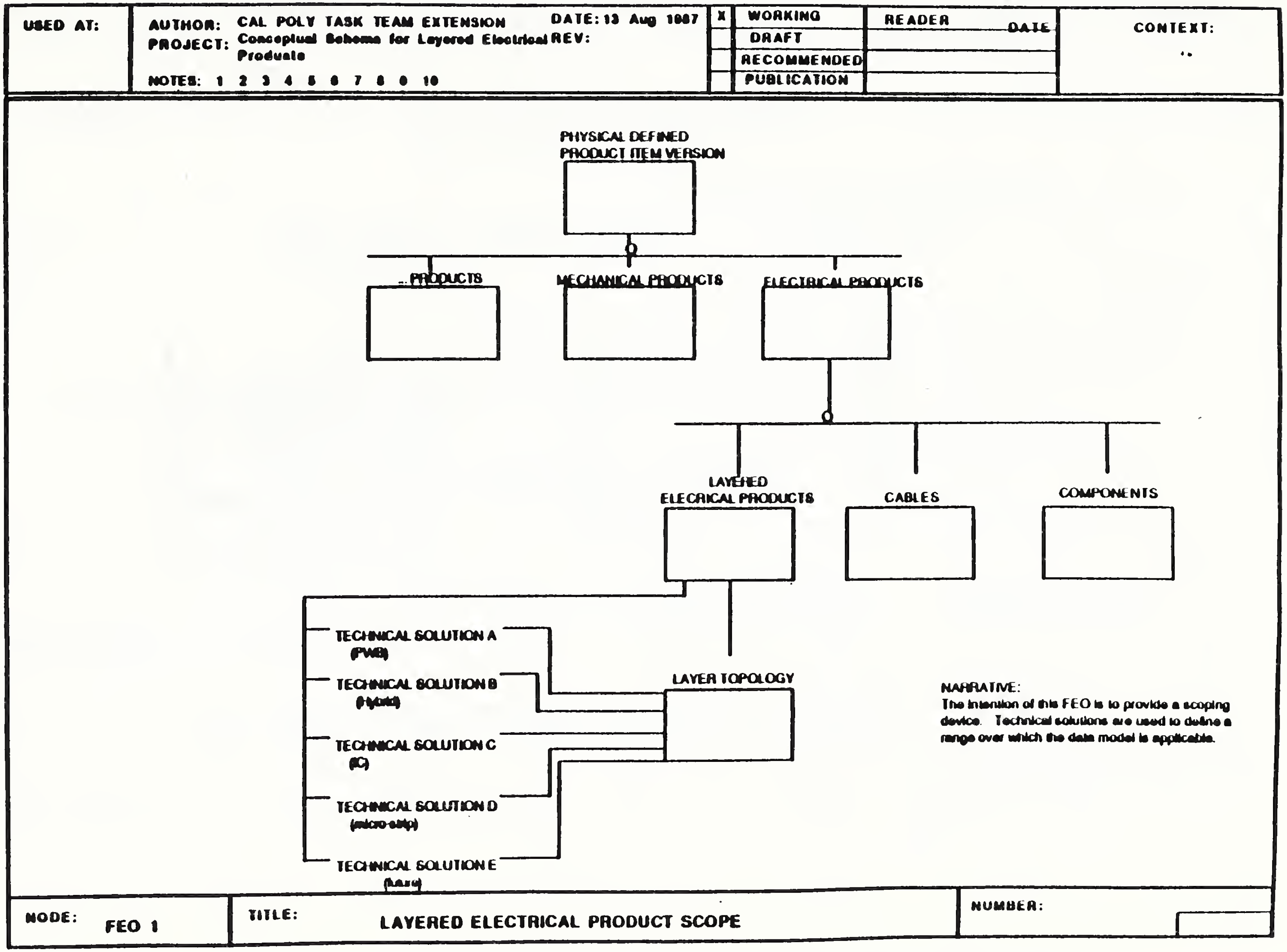
Figure 1. The Relationship of Functional Design Solutions, Physical Design Solutions and the Requirements Structure as defined by the Cal Poly Task Team, April 1987.



CIM DATABASE

Data stored in topological features, layer definition, detail mapping, holes, etc. in non-redundant object oriented form. CAD systems will support queries about pads and traces. Planning systems will support graphic, drill, route, and release B. O. M. queries. Any access to a CIM database can review file data, dictionary, coordinate, planning and test data.

Figure 2. The Computer Integrated Manufacturing Environment.


PHYSICAL DEFINED PRODUCT ITEM VERSION
... PRODUCTS
MECHANICAL PRODUCTS
ELECTRICAL PRODUCTS
LAYERED ELECRICAL PRODUCTS
CABLES
COMPONENTS
LAYER TOPOLOGY
TECHNICAL SOLUTION A (PWB)
TECHNICAL SOLUTION B (Hybrid)
TECHNICAL SOLUTION C (IC)
TECHNICAL SOLUTION D (micro-strip)
TECHNICAL SOLUTION E

USED AT:

AUTHOR: CAL POLY TASK TEAM EXTENSION **DATE:** 13 Aug 1987

PROJECT: Conceptual Schema for Layered Electrical Products **REV:**

NOTES: 1 2 3 4 5 6 7 8 9 10

☒ WORKING
☐ DRAFT
☐ RECOMMENDED
☐ PUBLICATION

READER **DATE**

CONTEXT:

NARRATIVE:
The intention of this FEO is to provide a scoping device. Technical solutions are used to define a range over which the data model is applicable.

NODE: FEO 1 **TITLE:** LAYERED ELECTRICAL PRODUCT SCOPE **NUMBER:**

Figure 3. The Range of Technical Solutions found in the Cal Poly Task Team Extension model.

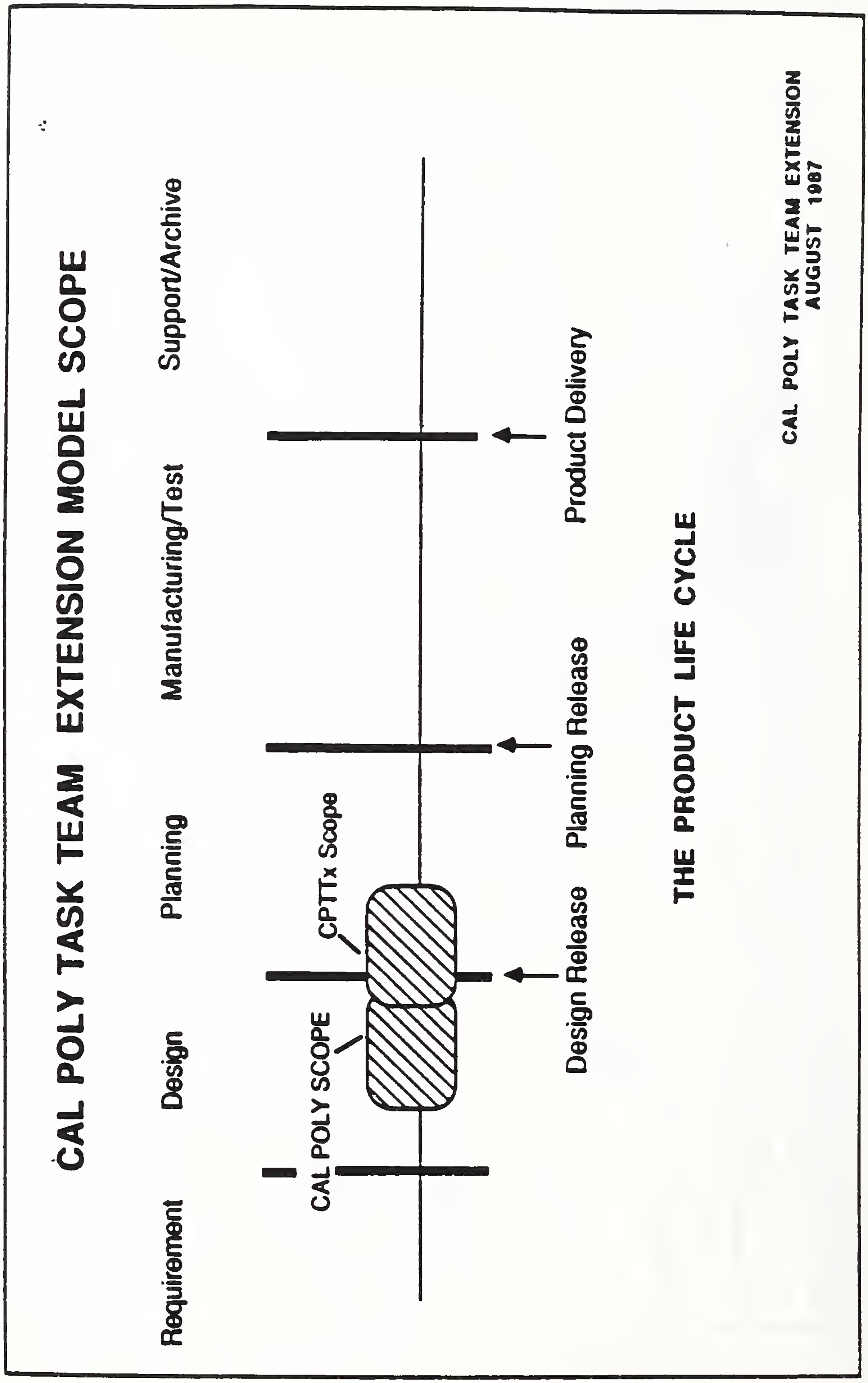


Figure 4. Scope of the Cal Poly Task Team Extension

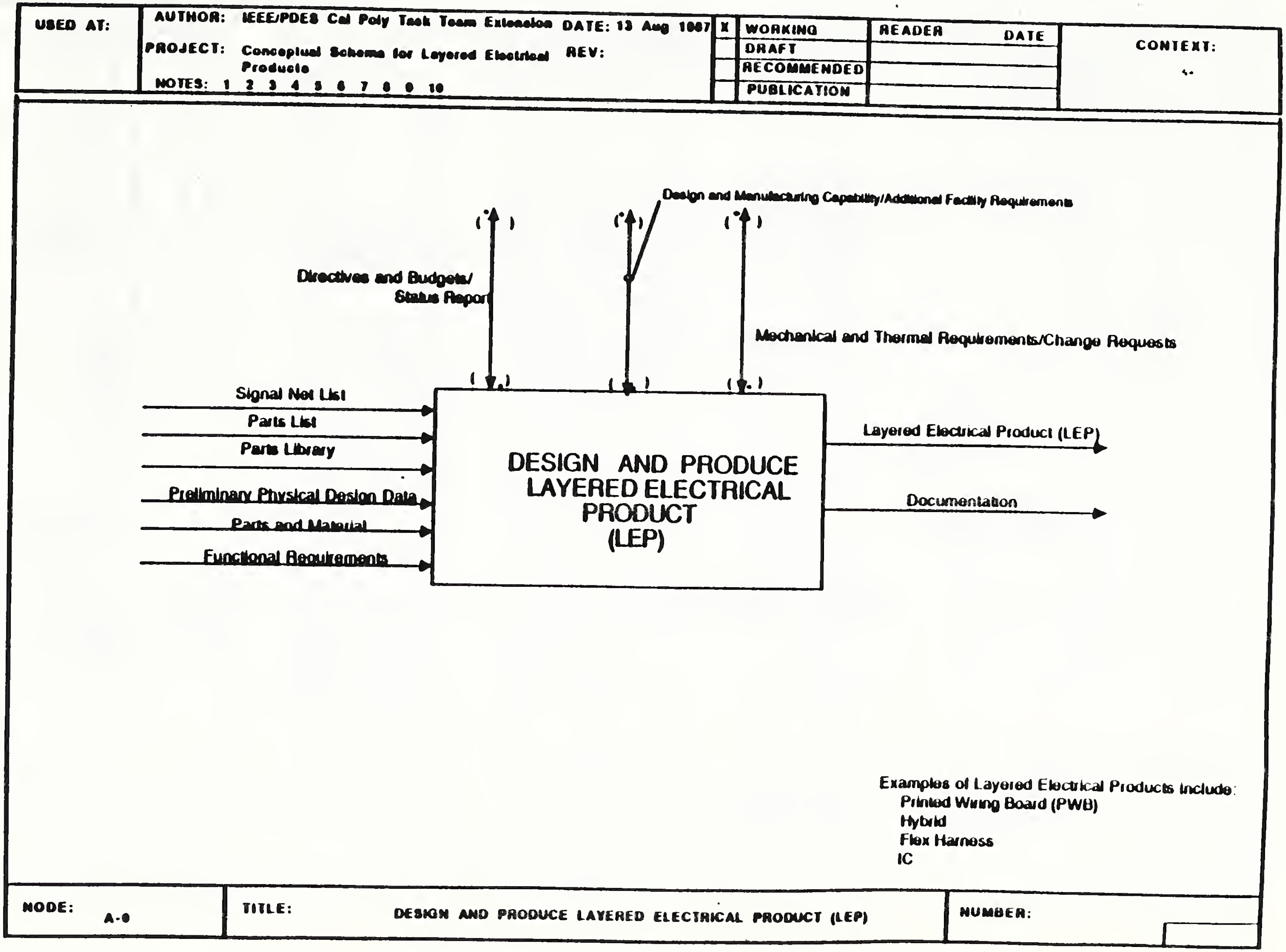

USED AT:
AUTHOR: IEEE/PDES Cal Poly Task Team Extension
DATE: 13 Aug 1987
PROJECT: Conceptual Schema for Layered Electrical Products
REV:
NOTES: 1 2 3 4 5 6 7 8 9 10
X
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
Design and Manufacturing Capability/Additional Facility Requirements
Directives and Budgets/
Status Report
Mechanical and Thermal Requirements/Change Requests
Signal Net List
Parts List
Parts Library
Preliminary Physical Design Data
Parts and Material
Functional Requirements
DESIGN AND PRODUCE LAYERED ELECTRICAL PRODUCT (LEP)
Layered Electrical Product (LEP)
Documentation
Examples of Layered Electrical Products include:
Printed Wiring Board (PWB)
Hybrid
Flex Harness
IC
NODE: A-0
TITLE: DESIGN AND PRODUCE LAYERED ELECTRICAL PRODUCT (LEP)
NUMBER:

Figure 5. A-0 Activity Diagram for the Cal Poly Task Team Extension model.

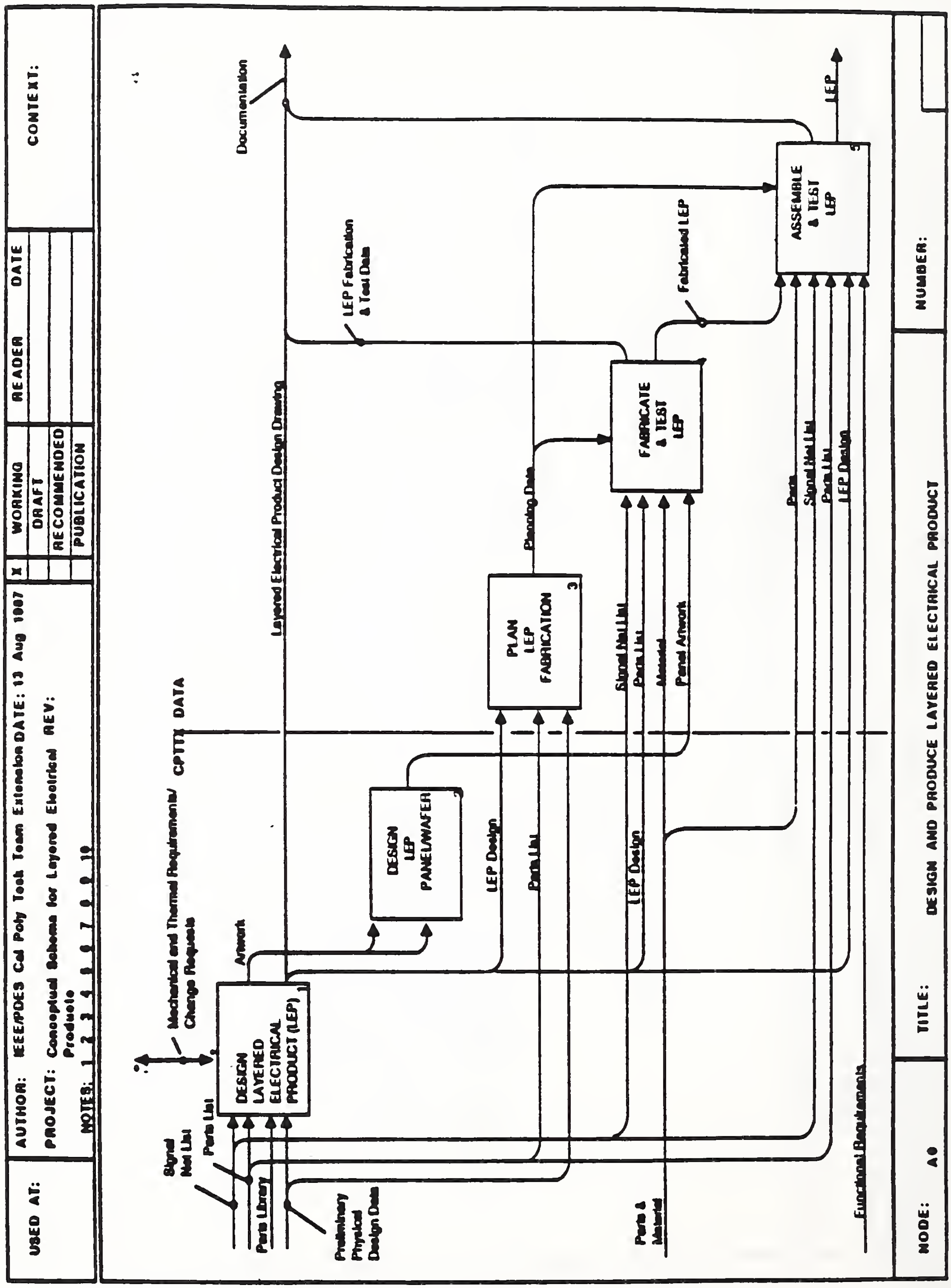


Figure 6. A0 Activity Diagram for the Cal Poly Task Team Extension model.

2.4 DEMONSTRATION OF THE MODEL

A prototype demonstration of the model was presented Wednesday January 13, 1988, during the meeting of the IGES/PDES General Assembly. This demonstration was designed to illustrate the suitability of the model in a production environment. The data for the board used in the demonstration test case was supplied by Westinghouse. It was also used by the Product Definition Data Interface (PDDI) project, a US Air Force CRAD contract carried out in conjunction with Westinghouse and McDonnell Aircraft. In the demonstration, several queries were run against a populated database. The database was constructured in DBase III. The queries were designed in Sequential Query Language (SQL) to answer popular user questions such as: 1) "Show the board outline", and 2) "What are all the traces on a given layer at a given voltage?" Answers to the queries were then graphically displayed using an interface program written by Dzung Ha. Appendices 8 and 9 contain sample database tables and examples of these queries.)

2.5 SELECTION OF A MODELING LANGUAGE

IDEF1X was selected as the modeling language for the project because it is the language used by consensus within the PDES community to develop conceptual models. The previous Cal Poly team had also used IDEF1X, and it was assumed that the next task would be to integrate these two electrical models. Because the Modeling Leader, John Zimmerman, is proficient in several modeling languages, we had an option to select from a variety of modelling languages without loss in productivity.

IDEF1X was developed by the U.S. Air Force ICAM project. A manual describing the modeling language and methodology is available from the ICAM-CIM Library at the Wright Aeronautical Laboratories in Dayton, Ohio. The team also received travel fund assistance through the Air Force Computer Integrated Manufacturing (CIM) Office, the successor to the ICAM program which had originally codified the IDEF modeling language and methodology in 1981. According to the ICAM specifications, IDEF 'kits' were prepared to record the results of each working session. These kits were sent to team members for review by individuals between meetings. At the beginning of each meeting these kits were collectively reviewed and revised to reflect consensus changes.

IDEF models were used with JANUS, a software package which outputs a normalized IDEF1X data model. This normalized model was used to check the team model in its final kit form. The input form for JANUS is a text language called Structural Modeling Language (SML). SML is a formal language which precisely describes each entity, attribute and relationship found in the model. It is intended to be concise for ease of entry and edit, and structured in formal syntax for computer readability. The SML input file describes each entity, the entity's parent(s), and the relation required to migrate primary and non-key attributes into the child entities.

The JANUS compiler checks SML syntax, migrates the keys and writes formal output reports. Those reports include the normalized SML, business rules, the entity and attribute glossary, and a graph of the model.

The JANUS software on site at the National Bureau of Standards, Factory Automation Systems Division (part of the AMRF) was used for the reconcillation of the input SML and the output description of the model. The SML input file and glossary were mutually loaded from the Team's composite view model. The AMRF staff successively ran the

JANUS normalization procedure, then compared the graphic model output to the source model. Relationships not in agreement were corrected by changing the input SML file.

2.6 RELATIONSHIP TO THE PDDI PROJECT

During the time when the Team was meeting, the PDDI project was also underway to determine data requirments for printed wiring (or circuit) boards. The PDDI Extension sought to apply the formal specification developed under the original PDDI Project to electronic products, specifically, circuit card assemblies. Although much of the subject data was the same, the approach and the results were different. PDDI wanted to apply the existing methodology to an electronic product; Cal Poly wanted to develop a rigorous entity-relationship definition for an electronic product. The Cal Poly Team received information, including the test data, from the PDDI Project, and other communication between the two groups was routinely exchanged.

2.7 SUMMARY OF THE WORKING SESSIONS

Although the setting and agenda varied for each meeting, there did seem to be a consistency of output. By the third session, after the scope and administrative issues settled down, the heart of the modeling activity began. The following is a brief meeting-by-meeting summary of the activity which occurred in each session.

SESSIONS I AND II: The Team utilized the first two meetings to collect source material and create a candidate entity pool. The source material consisted of prior electrical models from the IGES Electrical Application Committee and from the Air Force Product Definition Data Interface Extension Project. Definition specifications from the IEEE and the ANSI standard, ANSI/IPC-T-50C for Interconnecting and Packaging of Electronic Circuits (IPC) were also used. An IDEF1X kit of seven FEO's was assembled to define the model utilizing the entity pool which had been established.

SESSION III: The third meeting was the beginning of work on printed wiring board features. Our idea was to extend the PDES formal features model to provide for layered electrical products. The kit was revised to reflect this extension.

SESSION IV: During the fourth meeting, a review indicated that the modeling language was not serving as a sufficiently expressive vehicle to develop the concepts presented. A set of business rules was created. These rules replaced the appropriate kit FEO's as a temporary review mechanism.

Prior to the fifth meeting, Scott Madsen utilized a set of IDEF symbols on a CADDS III schematic capture system to construct a 37-entity model from the business rules and FEOs. (A C-size reproduction of the final version of this global model is attached to the back cover of this report.)

SESSION V: Topology and Reference Path Analysis were the main topics at this meeting. A methodology for query analysis was developed. The difference in approach from the PDDI project was noted.

SESSION VI: Although the agenda for the sixth meeting included populating the model, the model was not yet stable, particularly in the areas of topology, geometry and PWB panels. A lot of good modeling work as accomplished at this session. A complete shape/size model was incorporated and the relationship of PWB panels was further defined.

SESSION VII: This was an interim session conducted in China Lake, CA. At this meeting the model was 'blessed' by a number of visiting local reviewers. (It may have been cursed by a few.) Refinements were made in the model to accomodate logical subregions, and a preliminary population of the database was begun.

SESSION VIII: The structure of the Final Report, query capability, and questions on the demonstration project were discussed in detail at the last group session.

2.8 RESULTS

The Cal Poly Task Team Extension has produced a conceptual schema for all layered electrical products. By describing the fundamental elements about layered electrical products, the Task Team has paved the way for any number of user-specific external schemas to be built. (Note that only one technology-specific external schema (that for PWBs) has been shown.)

The Task Team Extension began its activities in the first week of May 1987 and completes its formal acitivity with the delivery of this final report. No further activity is automatically assumed by the Task Team.

2.9 FUTURE ACITIVITIES

The Model Glossary is being entered into a dictionary testbed. The testbed is built to the Information Resource Dictionary draft standard specification which is being developed by Helen Woods at the National Bureau of Standards. For further information on this activity, please see: A Technical Overview of the Information Resource Dictionary System (second edition) by Alan Goldfine and Patricia King.[1]

Further modeling tasks might include the development of an electronics component model and the integration of the functional and physical models. However, no definitive plans have been confirmed as of the date of this writing. Interested parties should contact the chairman of the IGES/PDES Electrical Applications Committee for more information regarding future modeling activities.

---

[1] This document is obtainable from the U.S. Department of Commerce, NATIONAL BUREAU of STANDARDS, Institute for Computer Science and Technology, Gaithersburg, MD, 20899. Document Number: NBS-IRS8-3700.

3. THE DATA MODEL

3.1 MODEL VIEWS

Each View of the model is a logical partition (that is, a logical distinct subset) which addresses some important concept found in the model. The global model was divided into 19 distinct subsets each of which is depicted by a unique view of the model. An index to these 19 views can be found in Table 1 (below). This index lists the subject of each view and the entities contained within this subject. Following the index are the 19 model views themselves.

THE MODEL VIEWS

VIEW	TITLE	ENTITIES
VIEW 1	Product Structure Context Diagram	S1, S2, S3, S8, C1
VIEW 2	Layer Structure	C1, C2, C3, C4, S4
VIEW 3	Layer Element	C3, C5, C7, C8, S5
VIEW 4	Layer Description	C2, C3, C14, C28, C32, C33
VIEW 5	Logical Subregion	C12, C14, C22, C23, C29
VIEW 6	Shape	C15, C16, C17, C32, C34, C35
VIEW 7	Edge	SS9, SS11, SS17
VIEW 8	Shape Topology	C14, C15, C16, C17, C32, C33 C34, C35, SS2, SS7, SS11, SS29
VIEW 9	Path/Loop	SS7, SS9, SS10, SS29, SS30 SS31, SS32
VIEW 10	Face	SS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8
VIEW 11	Curve Geometry	SS9, SS11, SS13, SS14, SS15, SS16, SS17, SS33
VIEW 12	Join Topology	C3, C5, C8, C9, C10, C18, SS9,SS11
VIEW 13	Boundary	C1, C2, C3, C5, C25, SS2
VIEW 14	External Shape	C17, C30, C31, SS2
VIEW 15	Component	S6, S7, S10, S11, P6, P10
VIEW 16	Deposition	C2, S4, S5, P6, P8, P9, P11, P16
VIEW 17	Signal	C3, C7, S5, P8, P10, P11, P12, P15
VIEW 18	Panel	C1, C2, P1, P2, P3, P4, P5, P6, S11
VIEW 19	Passage	C3, P6, P7, P10, P13, SS7, SS29
	Global Model	All

Table 1. Index to the Cal Poly Task Team Extension Model Views

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	X	WORKING	READER DATE	CONTEXT:... S1, S2, S3, S8, C1
				DRAFT		
				RECOMMENDED		
				PUBLICATION		

The four entities identified by dotted lines belong to the PDES Planning Model. The structure of these entities may change, but they suggest the idea that all products (parts and assemblies of parts) are instances of S1.

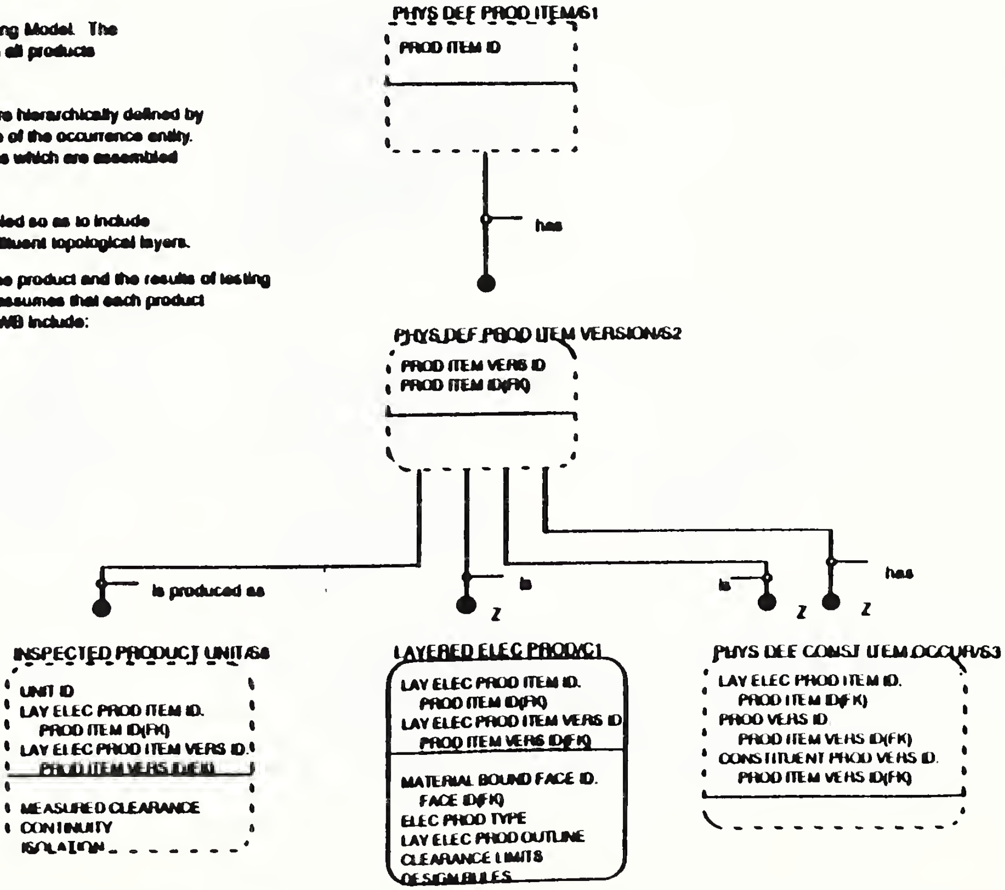
All products have at least one version identified in S2, and products are hierarchically defined by a pair-wise listing of product and product component as each instance of the occurrence entity. Thus for any level of assembly, these instances of S3 identify the items which are assembled into that product level.

The entity Layered Electrical Product is a "type of" product. It is modeled so as to include the physical design definition of any product defined in terms of constituent topological layers.

The Inspected Product Item entity contains data about instances of the product and the results of testing that instance to clearance limits specified for the product. This entity assumes that each product unit can be uniquely identified. The attributes it would have for the PWB include:

- annulus ring
- over-etch, under-etch

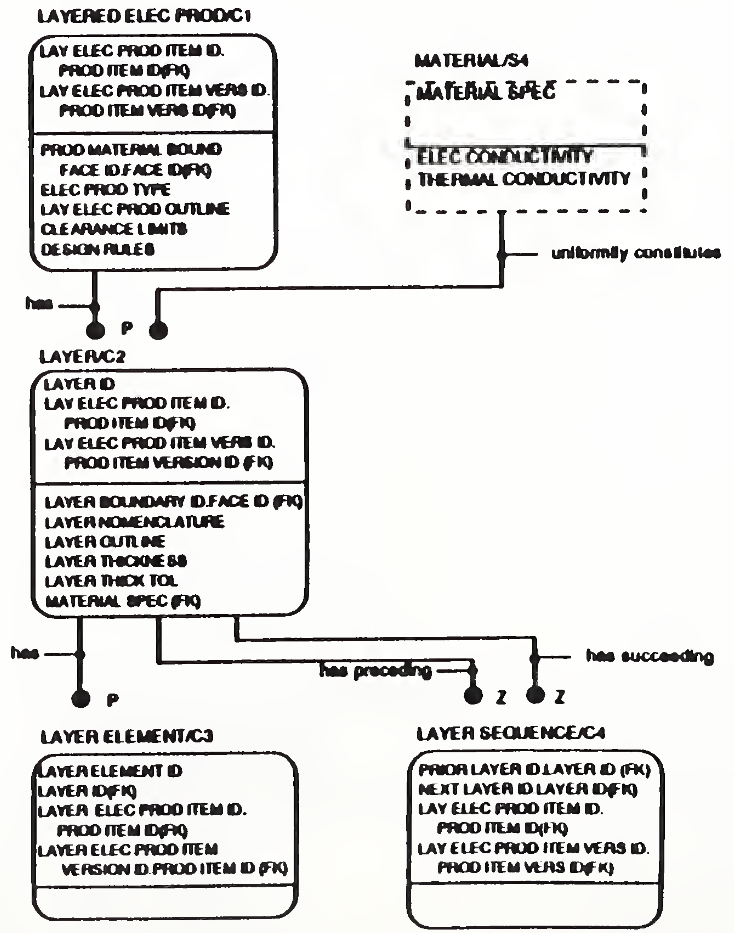
.
.
.



NODE: VIEW 1	TITLE: PRODUCT STRUCTURE CONTEXT DIAGRAM	NUMBER: A/ELEP/A

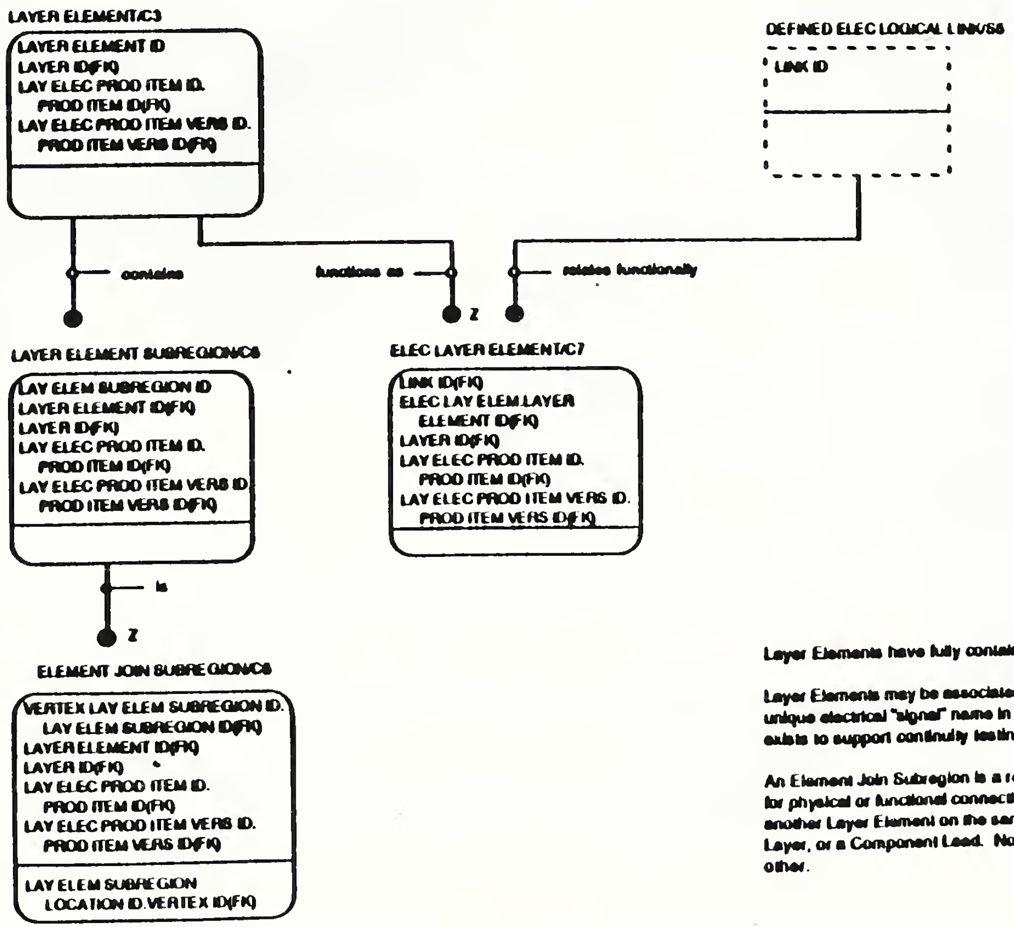
USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 28 Jan 1988 REV:	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE
CONTEXT: C1, C2, C3, C4, S4				

There are two very important ideas in this view. 1) A layer is constituted of only one material and 2) that it is contained in only one region. A layer is a 2-dimensional region that contains elements constituted of the same material. Layer elements may completely consume the region of the layer. Layers may contain voids. A Layered electrical product is a kind of electrical product that is built up of several layers. When multiple layers are involved, some sort of ordering is implied. Layer sequence defines the preceding and succeeding sequence for a given layer.



NODE: VIEW 2	TITLE: LAYER STRUCTURE	NUMBER: A/ELEP/A

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 28 Jan 1988 REV:	X	WORKING	READER	DATE	CONTEXT:
				DRAFT			C3, C5, C7, C8, S5
				RECOMMENDED			
				PUBLICATION			



Layer Elements have fully contained subregions.

Layer Elements may be associated with an electrical signal. By relating a unique electrical "signal" name in the Electrical Layer Element, the data exists to support continuity testing.

An Element Join Subregion is a region of a Layer Element that is intended for physical or functional connection. A Layer Element may connect with another Layer Element on the same Layer, a Layer Element on a different Layer, or a Component Lead. Note that Layer Elements do not touch each other.

NODE: VIEW 3	TITLE: LAYER ELEMENT	NUMBER: A/ELEP/A

USED AT:
AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE: 28 Jan 1988
PROJECT: Conceptual Schema for Layered Electrical Products REV:
NOTES: 1 2 3 4 5 6 7 8 9 10

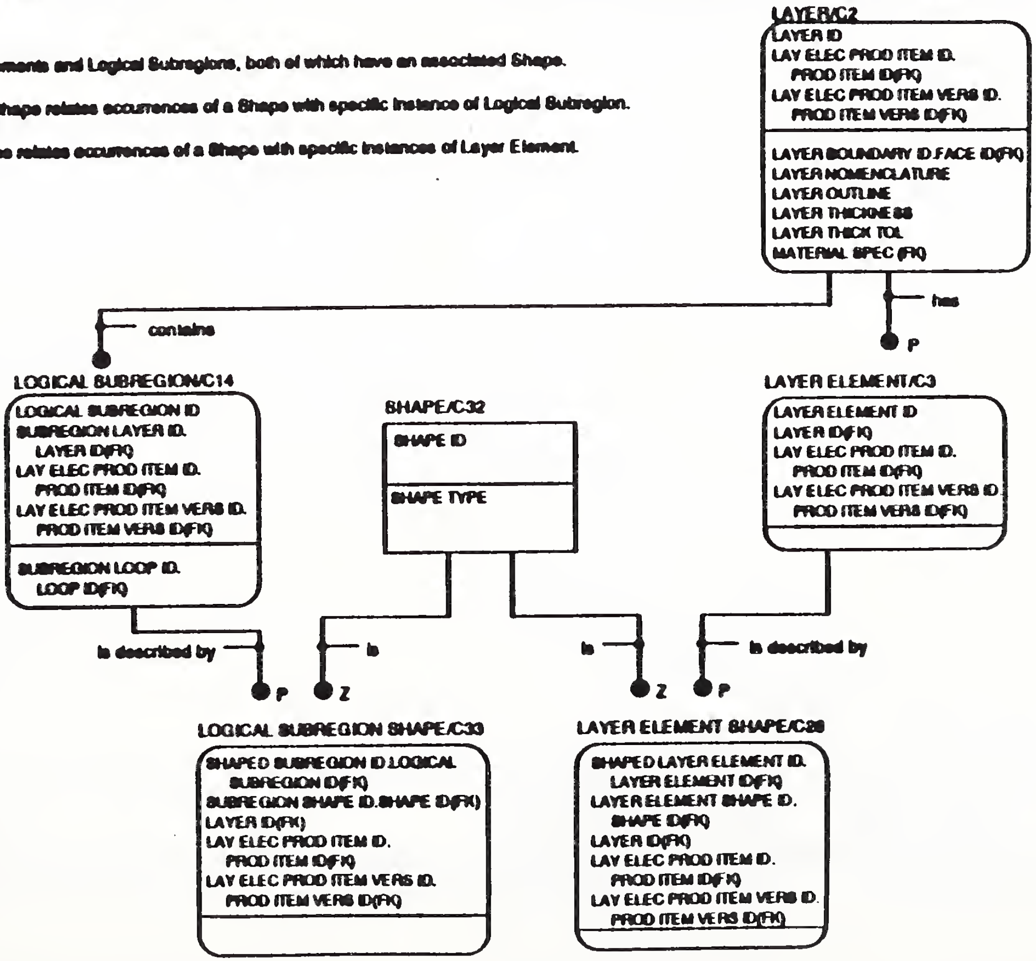
		READER	DATE
X	WORKING		
	DRAFT		
	RECOMMENDED		
	PUBLICATION		

CONTEXT:
C2, C3, C14, C28, C32, C33

Layers contain Layer Elements and Logical Subregions, both of which have an associated Shape.

The Logical Subregion Shape relates occurrences of a Shape with specific instance of Logical Subregion.

The Layer Element Shape relates occurrences of a Shape with specific instances of Layer Element.



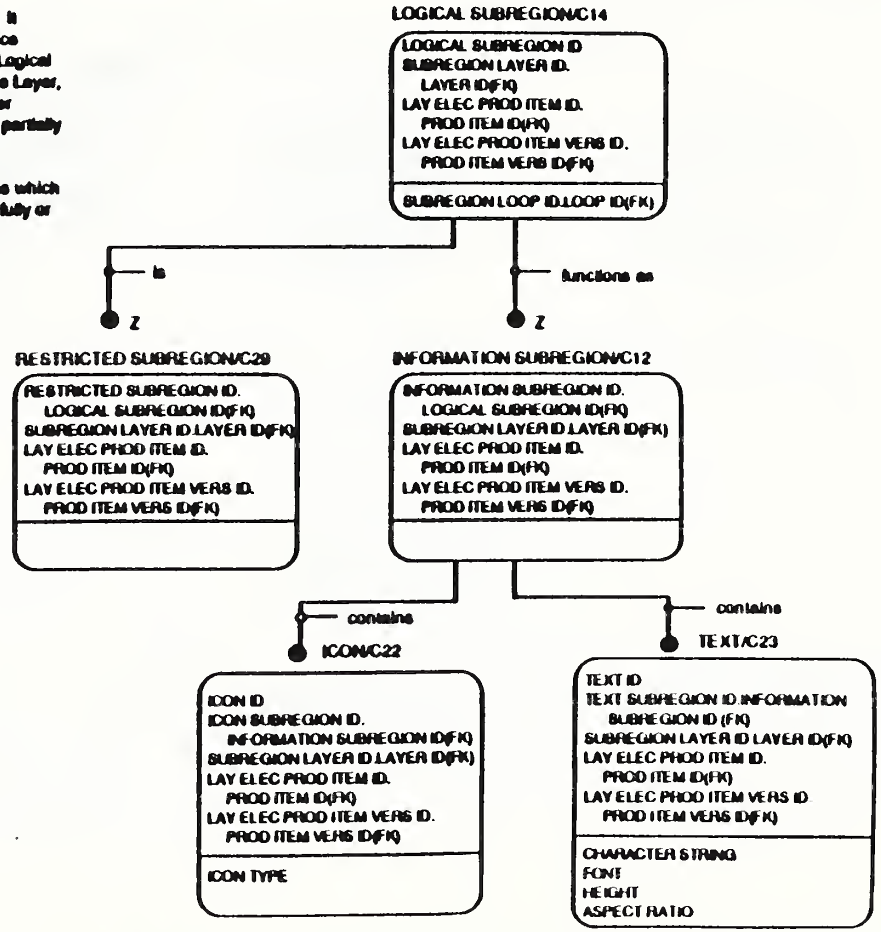
NODE: VIEW 4
TITLE: LAYER DESCRIPTION
NUMBER: A/ELEP/A

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 28 Jan 1988 REV:	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT: C12, C14, C22, C23, C29

Note that a Logical Subregion does not have to contain any Material. It only has to be within the region of the Layer which is a major difference from the definition of Layer Element which must contain Material. A Logical Subregion is an artificial area which is superimposed anywhere on the Layer, for example, on top of an Layer Element, crossing over a boundary, or superimposed over a void. Note that one Logical Subregion may be partially or fully contained within another Logical Subregion.

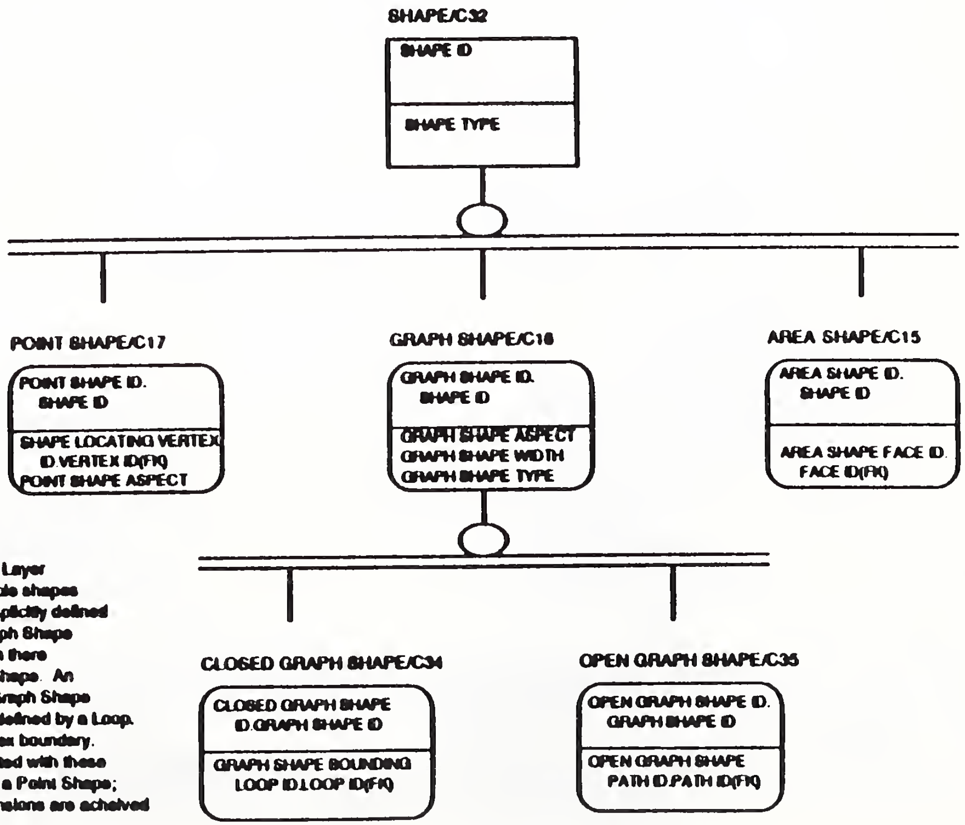
Special types of Logical Subregions include: Information Subregions which contain Icons and Text, and Restricted Subregions which are either fully or partially defined keep-out boundaries.

Icon Type attribute may be:
  Registration Mark
  Part Polarity
  Company Logo
  Bar Code
  Component Outline



NODE: VIEW 8	TITLE: LOGICAL SUBREGION	NUMBER: A/ELEP/A

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	X	WORKING	READER	DATE	CONTEXT: C15, C16, C17, C32, C34, C35
				DRAFT			
				RECOMMENDED			
				PUBLICATION			



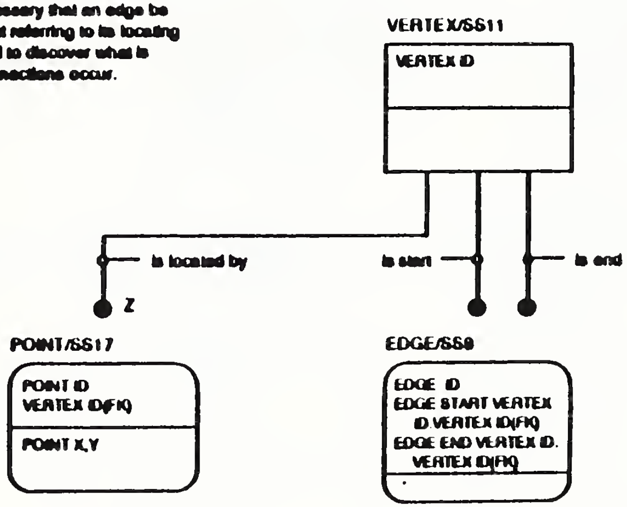
Four basic Shapes are used to describe the outline of a Layer Element or a Logical Subregion. Point Shapes are simple shapes located by a single Point, for example, a pad which is implicitly defined by a position on a photo plotter apperture wheel. A Graph Shape is a simple linear shape that has a uniform width of which there are two types: Open Graph Shape and Closed Graph Shape. An example of an Open Graph Shape is a trace; a Closed Graph Shape is an irregular area, like a ground plane region which is defined by a Loop. An Area Shape is a region which has an arbitrary complex boundary. Ascending topological degrees of freedom are associated with these geometric shapes: the zero dimension corresponds to a Point Shape; one dimension is represented by a Line, and two-dimensions are acheived by the depiction of an Area Shape.

Graph shapes are not allowed to have branches. Such branches must be modeled as multiple connected graph shapes.

NODE: VIEW 6	TITLE: SHAPE	NUMBER: A/ELEP/A

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE:8 Jan 1988 PROJECT: Conceptual Schema for Layered Electrical Products REV: NOTES: 1 2 3 4 5 6 7 8 9 10	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT: SS9, SS11, SS17

A vertex is an abstraction for some object to be connected. A vertex can play the role of being either a start or a stop vertex for an edge or it can play both roles, for example, when a single edge loops on a single vertex. When a vertex is associated with more than one edge, it topologically connects those edges. An edge is associated with exactly one start vertex and one stop vertex. A vertex is said to be located by a point but it is not necessary that an edge be located by a point. One can refer to the vertex without referring to its locating point. This is done often in queries on the LEP model to discover what is connected to what without finding out where the connections occur.



NODE: VIEW 7	TITLE: EDGE	NUMBER:

USED AT:

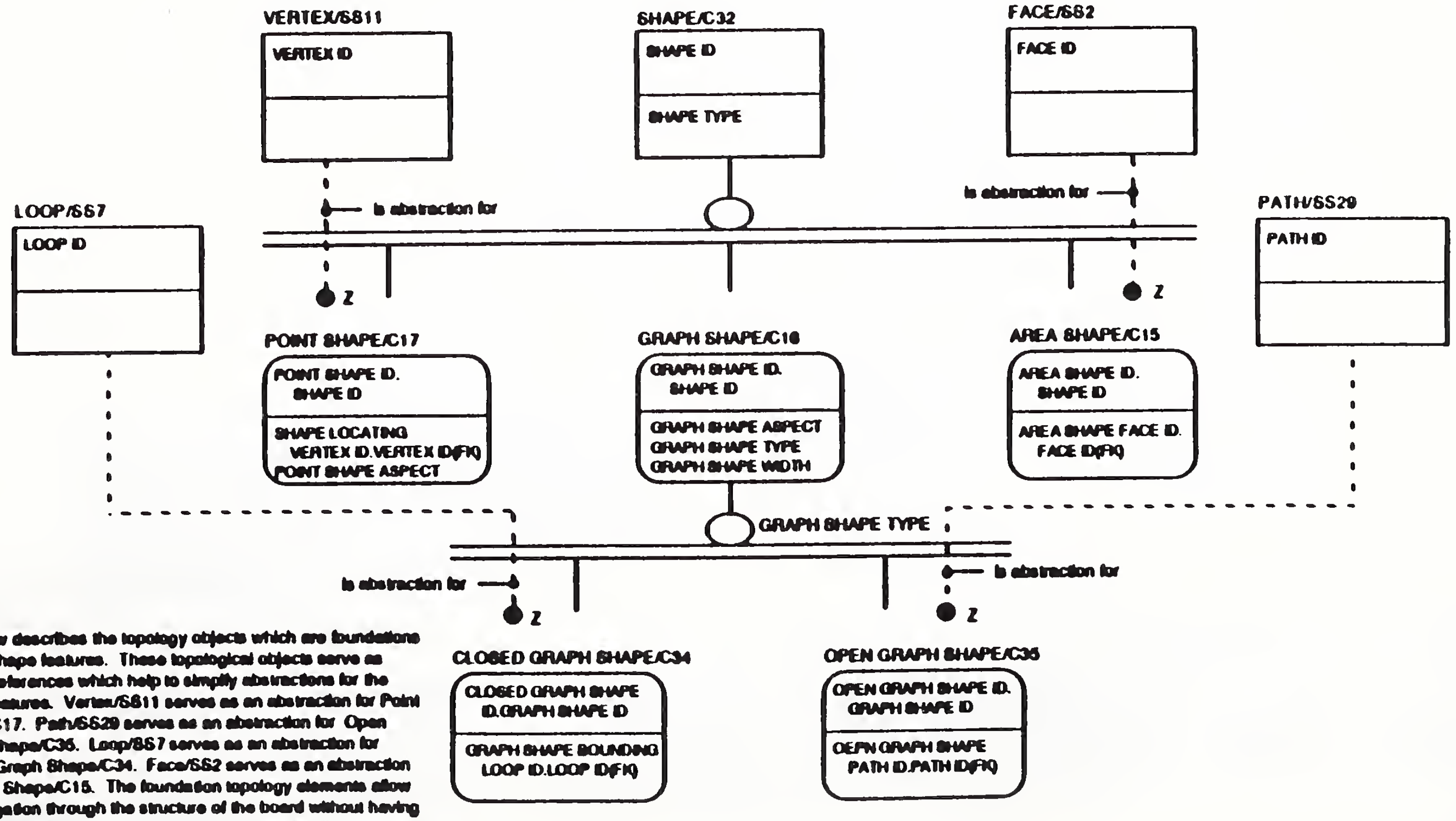
AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE: 8 Jan 1988
PROJECT: Conceptual Schema for Layered Electrical Products REV:

NOTES: 1 2 3 4 5 6 7 8 9 10

☒ WORKING
☐ DRAFT
☐ RECOMMENDED
☐ PUBLICATION

READER DATE

CONTEXT:
C14, C15, C16, C17,
C32, C33, C34, C35,
SS2, SS7, SS11, SS29



This view describes the topology objects which are foundations for the shape features. These topological objects serve as mutual references which help to simplify abstractions for the shape features. Vertex/SS11 serves as an abstraction for Point Shape/C17. Path/SS29 serves as an abstraction for Open Graph Shape/C35. Loop/SS7 serves as an abstraction for Closed Graph Shape/C34. Face/SS2 serves as an abstraction for Area Shape/C15. The foundation topology elements allow for navigation through the structure of the board without having to refer to geometry. It brings a level of standardization to the shape model. Topology is a well understood and mathematically rigorous system that is the basis for many formal systems that deal with connected objects.

NODE: VIEW 6 TITLE: SHAPE TOPOLOGY NUMBER:

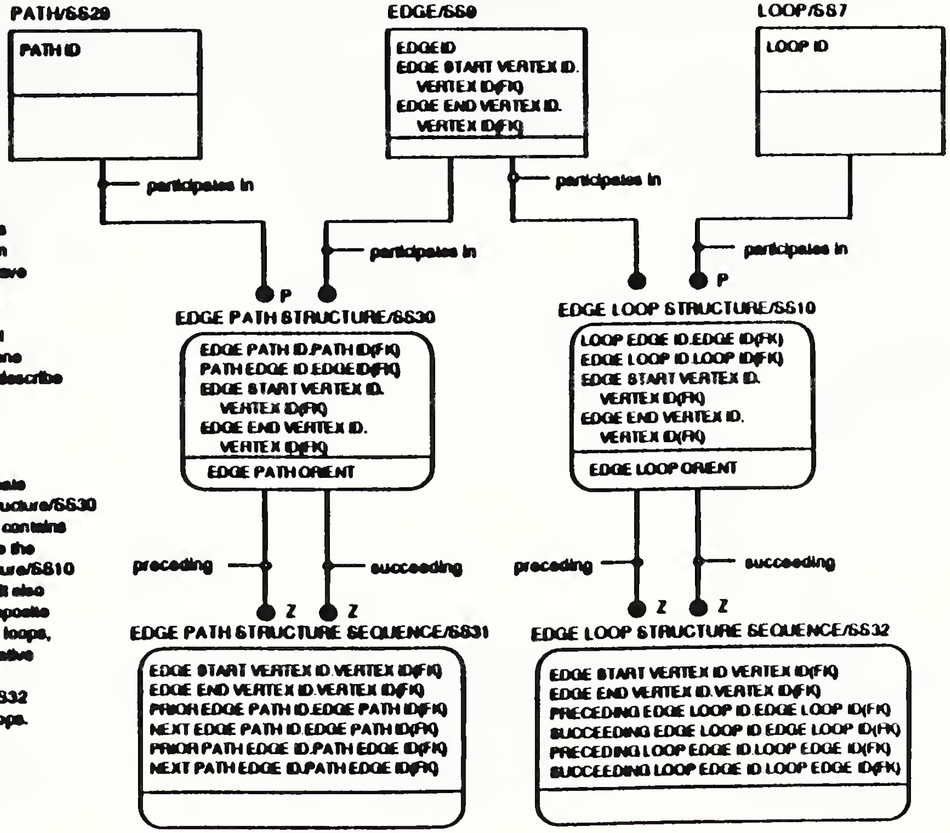
USED AT:

AUTHOR: IEEE/PDES Cal Poly Task Team Extension
PROJECT: Conceptual Schema for Layered Electrical Products
DATE: 8 Jan 1988
REV:
NOTES: 1 2 3 4 5 6 7 8 9 10

		READER	DATE
X	WORKING		
	DRAFT		
	RECOMMENDED		
	PUBLICATION		

CONTEXT: SS7, SS9, SS10, SS29, SS30, SS31, SS32

One of the major roles of the edge in the topology system is to participate in loops and paths. The edge is a basic building block for loops and paths. In turn, paths and loops are basic building blocks in the LEP model. Paths are open strings of connected edges (the first and last edge must have a vertex that is associated with only one edge of the path). Edges cannot have branches in them, so often times it is necessary to tie several edges together to get shapes that we want. An example of this is a trace on a PWB that has one or more branches. In this case we use the set of paths to describe the 'skeleton' of a trace. Loops are closed strings of connected edges (the first vertex of the first edge is the same as the last vertex of the last edge). Loops are necessary components in faces. We use faces to define the areas of explicit pads and layers. An edge may participate in both a loop and a path at the same time. Edge Path Structure/SS30 is an intersection entity which relates an edge to a path. It contains information that indicates whether the edge and path have the same or opposite directions. Similarly, Edge Loop Structure/SS10 is an intersection entity which relates an Edge to a Loop. It also indicates whether the edge and loop have the same or opposite directions. Edge Loop Structure/SS10 flags two types of loops, positive loops that bound the perimeter of a face and negative loops that create voids in the face. Edge Path Structure Sequence/SS31 and Edge Loop Structure Sequence/SS32 specify the order in which edges are used in paths and loops.

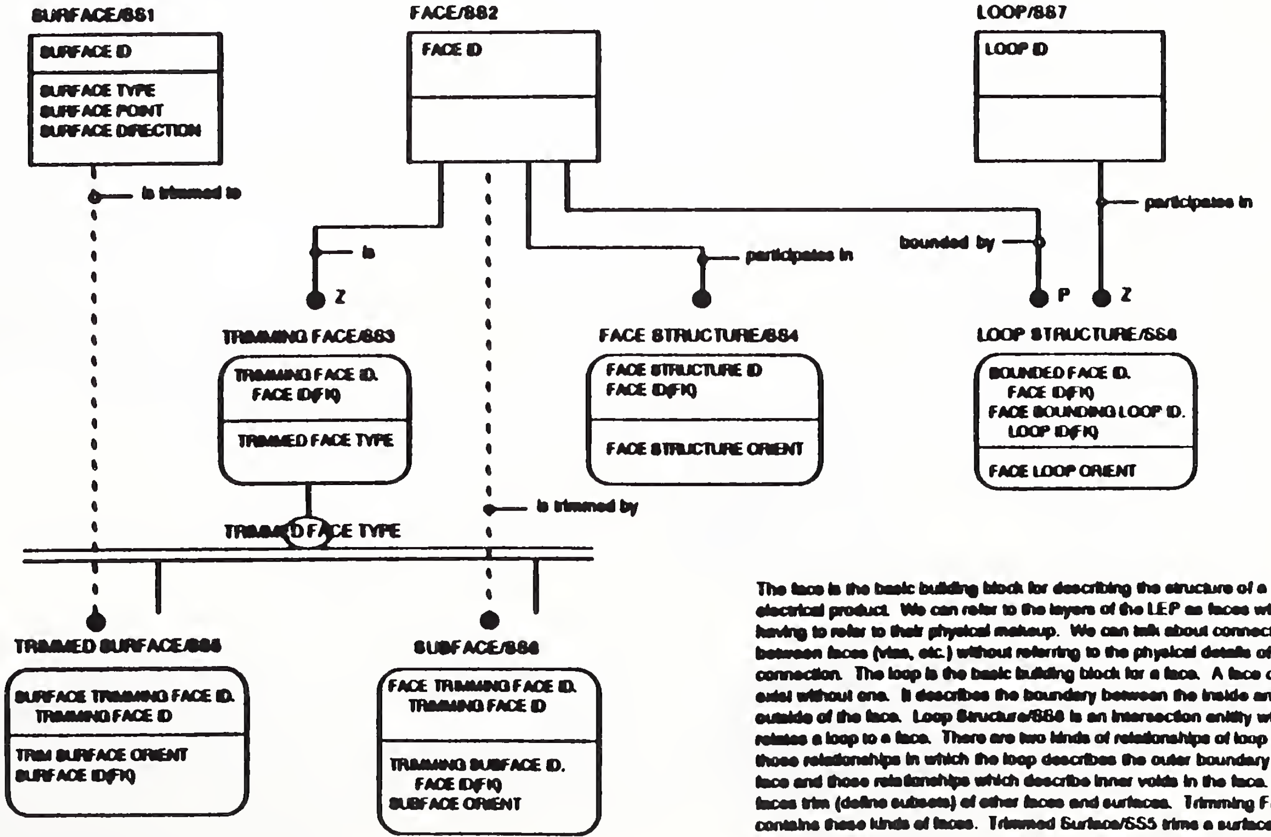

PATH/SS29
PATH ID
EDGE/SS9
EDGE ID
EDGE START VERTEX ID. VERTEX ID(FK)
EDGE END VERTEX ID. VERTEX ID(FK)
LOOP/SS7
LOOP ID
participates in
participates in
participates in
participates in
P
P
EDGE PATH STRUCTURE/SS30
EDGE PATH ID.PATH ID(FK)
PATH EDGE ID.EDGE ID(FK)
EDGE START VERTEX ID. VERTEX ID(FK)
EDGE END VERTEX ID. VERTEX ID(FK)
EDGE PATH ORIENT
EDGE LOOP STRUCTURE/SS10
LOOP EDGE ID.EDGE ID(FK)
EDGE LOOP ID.LOOP ID(FK)
EDGE START VERTEX ID. VERTEX ID(FK)
EDGE END VERTEX ID. VERTEX ID(FK)
EDGE LOOP ORIENT
preceding
succeeding
preceding
succeeding
Z
Z
Z
Z
EDGE PATH STRUCTURE SEQUENCE/SS31
EDGE START VERTEX ID.VERTEX ID(FK)
EDGE END VERTEX ID.VERTEX ID(FK)
PRIOR EDGE PATH ID.EDGE PATH ID(FK)
NEXT EDGE PATH ID.EDGE PATH ID(FK)
PRIOR PATH EDGE ID.PATH EDGE ID(FK)
NEXT PATH EDGE ID.PATH EDGE ID(FK)
EDGE LOOP STRUCTURE SEQUENCE/SS32
EDGE START VERTEX ID VERTEX ID(FK)
EDGE END VERTEX ID.VERTEX ID(FK)
PRECEDING EDGE LOOP ID.EDGE LOOP ID(FK)
SUCCEEDING EDGE LOOP ID EDGE LOOP ID(FK)
PRECEDING LOOP EDGE ID.LOOP EDGE ID(FK)
SUCCEEDING LOOP EDGE ID LOOP EDGE ID(FK)

NODE: VIEW 9
TITLE: PATH / LOOP
NUMBER:

USED AT:
AUTHOR: IEEE/PDES Cal Poly Task Team Extension
PROJECT: Conceptual Schema for Layered Electrical Products
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 8 Jan 1988
REV:

		READER	DATE
X	WORKING		
	DRAFT		
	RECOMMENDED		
	PUBLICATION		

CONTEXT:
SS1, SS2, SS3, SS4,
SS5, SS6, SS7, SS8

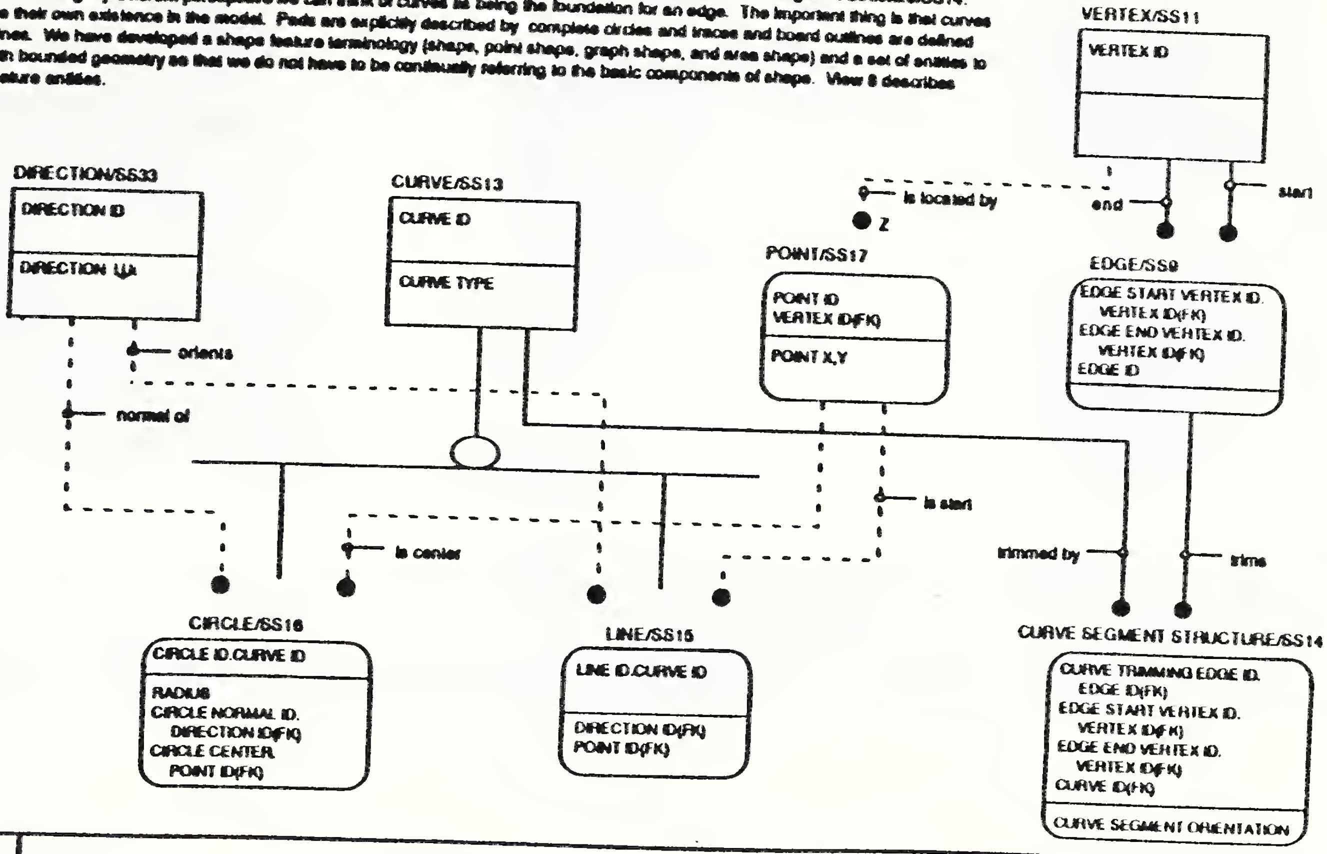


The face is the basic building block for describing the structure of a layered electrical product. We can refer to the layers of the LEP as faces without having to refer to their physical makeup. We can talk about connections between faces (vias, etc.) without referring to the physical details of the connection. The loop is the basic building block for a face. A face cannot exist without one. It describes the boundary between the inside and the outside of the face. Loop Structure/SS8 is an intersection entity which relates a loop to a face. There are two kinds of relationships of loop to face - those relationships in which the loop describes the outer boundary of the face and those relationships which describe inner voids in the face. Some faces trim (define subsets) of other faces and surfaces. Trimming Face/SS3 contains these kinds of faces. Trimmed Surface/SS5 trims a surface. Subface/SS6 trims another face. In the LEP model all surfaces are planes.

NODE: VIEW 10
TITLE: FACE
NUMBER:

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	x WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT: SS8, SS11, SS13, SS14, SS15, SS16, SS17, SS33

Only two kinds of curves are used in the LEP model. These are the entities, Circle/SS16 and Line/SS15. This view does not actually define
circle and line (these are undefined in our model), but rather indicate the how to construct them. A circle requries a Point/SS17 (center)
and a Direction/SS33 and a radius in order to be constructed. A line requires a Point/SS17 (start) and a Direction/SS33 in order to be constructed.
In our model all lines are unbounded and all circles are complete. However, by using Edge/SS9, we can define a portion of an unbounded line
or a portion of a full circle. This is done by relating Edge/SS9 to Curve/SS13 through the intersection entity Curve Segment Structure/SS14.
Looking at it from at a slightly different perspective we can think of curves as being the foundation for an edge. The important thing is that curves
and edges have their own existence in the model. Pads are explicitly described by complete circles and traces and board outlines are defined
by portions of lines. We have developed a shape feature terminology (shape, point shape, graph shape, and area shape) and a set of entities to
help us deal with bounded geometry so that we do not have to be continually referring to the basic components of shape. View 8 describes
these shape feature entities.



NODE: VIEW 11	TITLE: CURVE GEOMETRY	NUMBER:

USED AT:
AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE: 8 Jan 1988
PROJECT: Conceptual Schema for Layered Electrical Products REV:
NOTES: 1 2 3 4 5 6 7 8 9 10

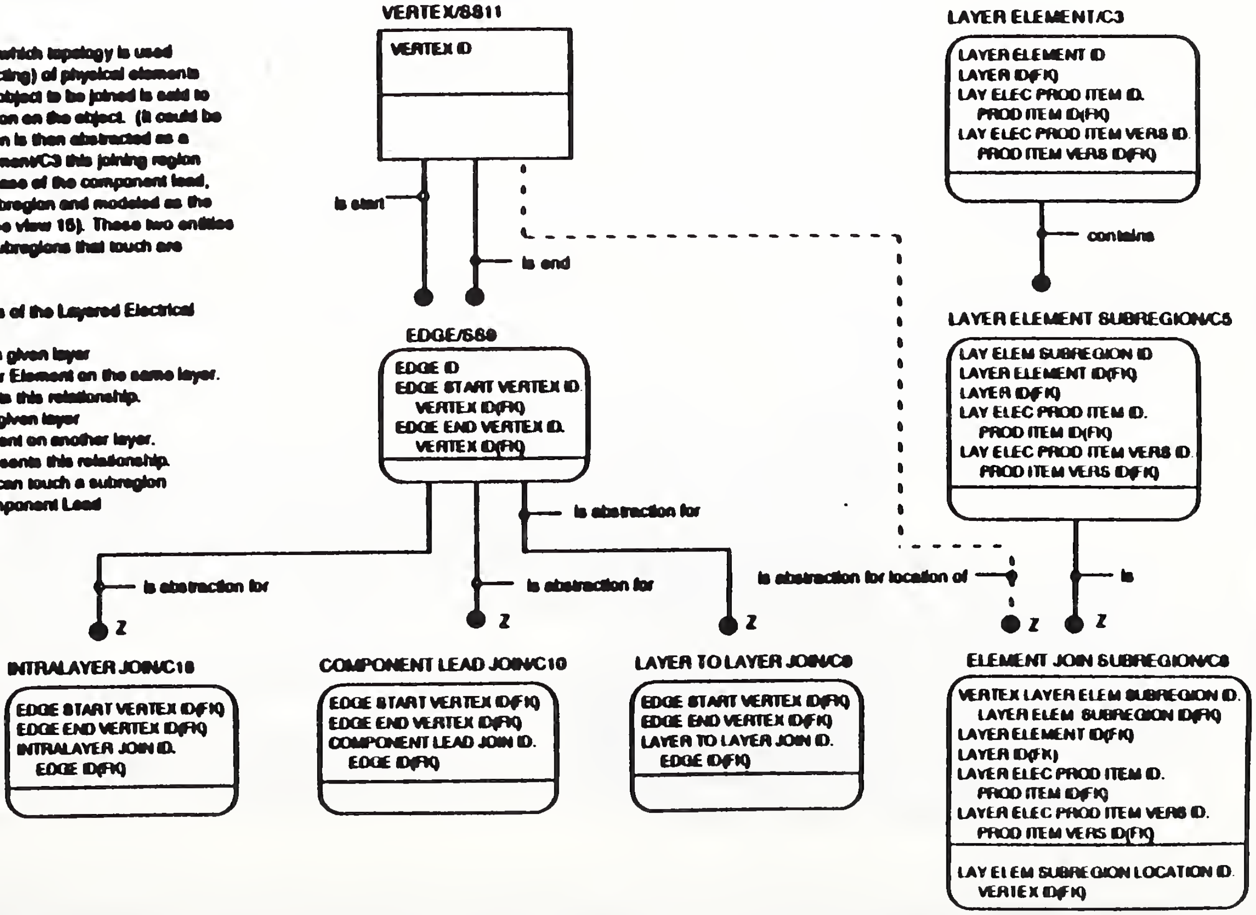
☒ WORKING
☐ DRAFT
☐ RECOMMENDED
☐ PUBLICATION

READER DATE

CONTEXT: C3, C5, C6, C9, C10, C18, SS9, SS11

This view is an exposition of the way in which topology is used to describe the joining (physical connecting) of physical elements of the layered electrical product. Each object to be joined is said to have a joining region which is a subregion on the object. (It could be the entire object.) That joining subregion is then abstracted as a single vertex. In the case of Layer Element/C3 this joining region is Element Join Subregion/C6. In the case of the component lead, the entire lead is treated as a joining subregion and modeled as the entity Component Lead Vertex/S10 (see view 16). These two entities are abstracted as Vertex/SS11. Two subregions that touch are abstracted as a single Edge/SS9.

Here are the various ways that elements of the Layered Electrical Product can touch:
1) A subregion on a Layer Element on a given layer can touch a subregion on another Layer Element on the same layer. The entity Intralayer Join/C18 represents this relationship.
2) A subregion on a layer element on a given layer can touch a subregion on a Layer Element on another layer. The entity Layer to Layer Join/C9 represents this relationship.
3) A subregion on a Component Lead can touch a subregion of some layer element. The entity Component Lead Join/C10 represents this relationship.



NODE: VIEW 12
TITLE: JOIN TOPOLOGY
NUMBER:

USED AT:

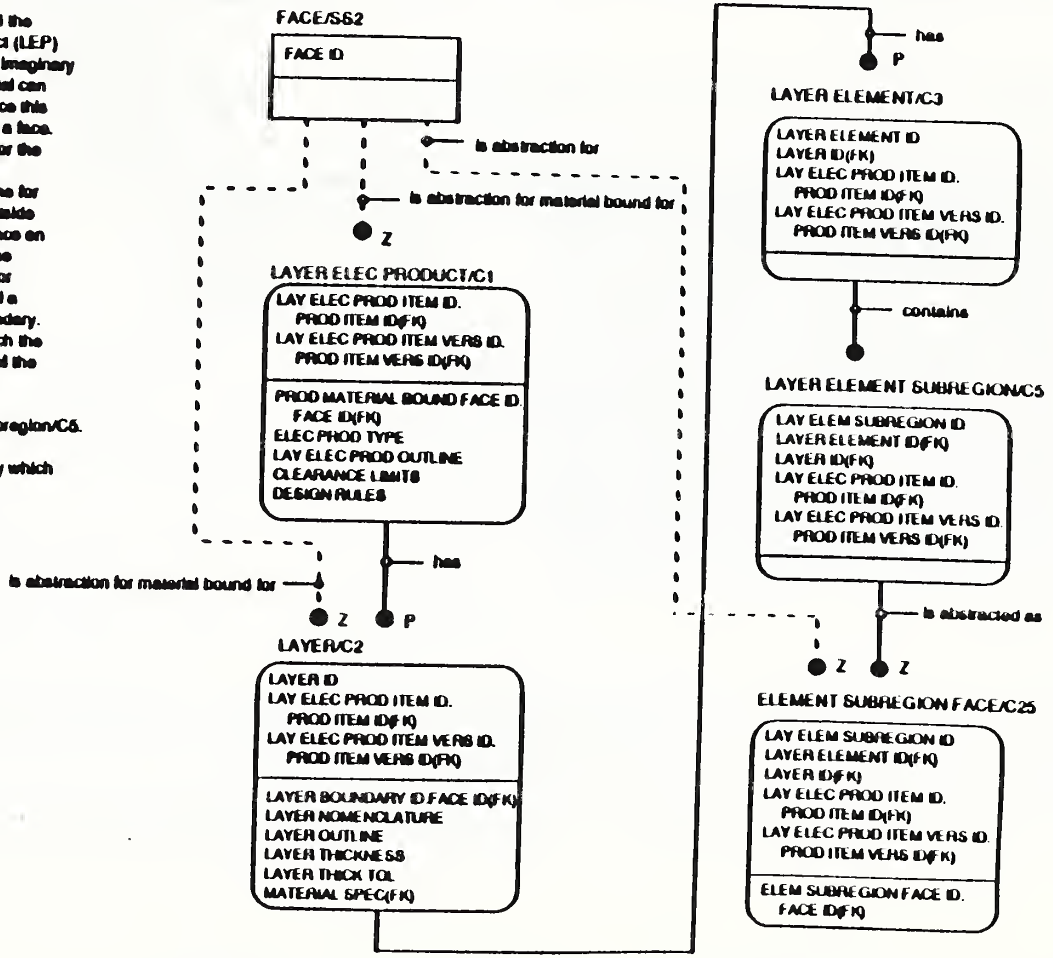
AUTHOR: IEEE/PDES Cal Poly Task Team Extension
PROJECT: Conceptual Schema for Layered Electrical Products
DATE: 8 Jan 1988
REV:
NOTES: 1 2 3 4 5 6 7 8 9 10

		READER	DATE	CONTEXT:
X	WORKING			C1, C2, C3, C5, C25, SS2
	DRAFT			
	RECOMMENDED			
	PUBLICATION			

This view describes how the topology entities model the material bound of the entire layered electrical product (LEP) and its individual layers. The material boundary is an imaginary line drawn to describe a region inside of which material can exist and outside of which material cannot exist. Since this line is describing inside-outside, it must be a loop for a face. (See topology tutorial.) Face/SS2 is an abstraction for the material bound for Layer Electrical Product /C1 and for Layer/C2. These boundaries may be the same for the LEP and its layers. No layer boundary can go outside the LEP boundary. A layer may have only a single trace on it - no parts of which touch the material boundary. The material boundary of that layer defines the potential for material. For example, if the design is changed to add a new trace, that trace must be inside the material boundary. It is not required that that any portion of any layer touch the boundary, however, it is possible that the perimeter of the layer is identical to this boundary.

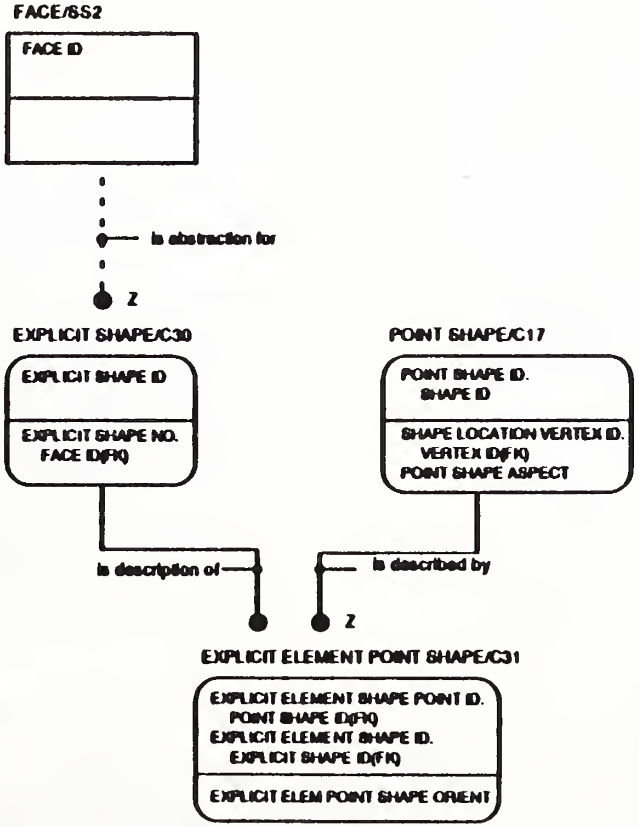
Face/SS2 is also used to abstract Layer Element Subregion/C5.

Element Subregion Face/C25 is an intersection entity which relates one layer element subregion to one face.



NODE: VIEW 13
TITLE: BOUNDARY
NUMBER:

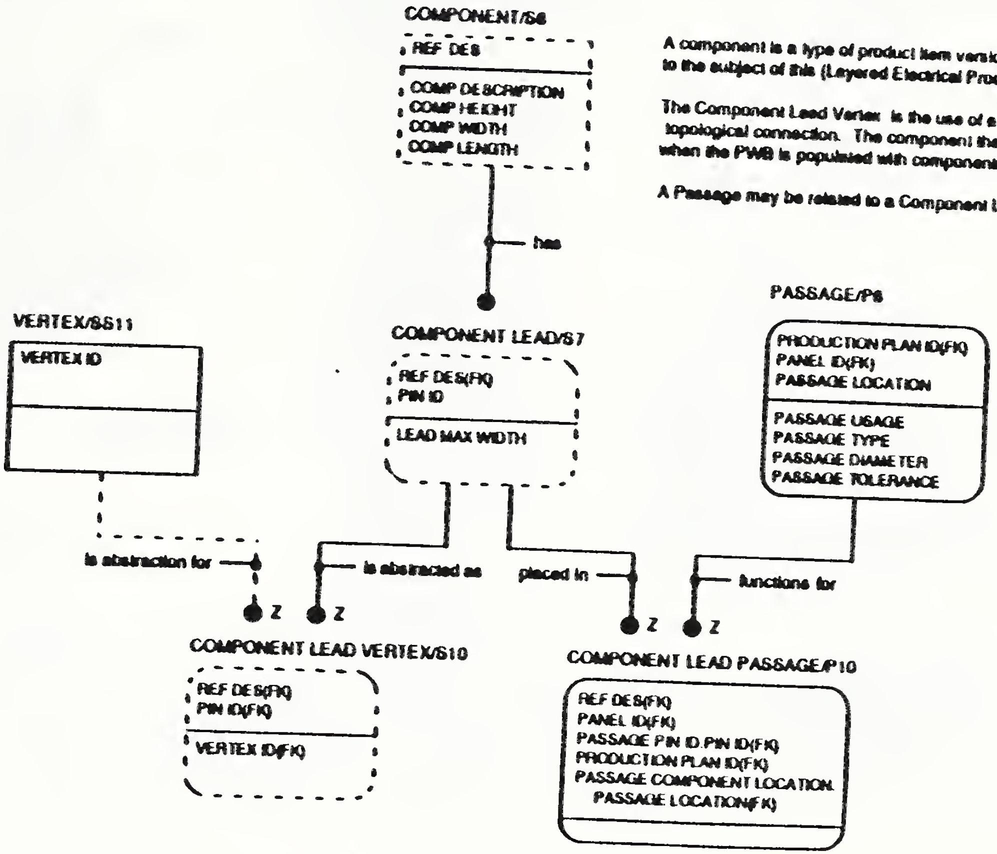
USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products	DATE: 6 Jan 1988 REV:	X	WORKING	READER	DATE	CONTEXT: C17, C30, C31, SS2
				DRAFT			
				RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION			



Point Shapes are often repititous on an LEP, for example, a round pad. In our LEP model, a point shape is modeled once explicitly as Explicit Shape/C30 and then instanced several times as a Point Shape/C17. Explicit Element Point Shape/C31 is an intersection record that relates a single explicit shape to a point shape. Although most point shapes are simple, this explicit shape may have an arbitrarily complex outline. It was desired in this model not to attempt to enumerate all possible point shapes (for example, flashes) but rather to allow the user of the model to create any explicit shape. One must follow the topological model through to determine the geometry associated with the explicit shape. The advantage in this model is that a standard topology model is used to describe the point shape. Point shapes can have voids. The voids are included in the shape as negative loops in the face. This is how holes in pads are modeled. Point shapes are normally considered to have a single void placed at the centroid of the shape but the model does not force this. Of particular interest in our model is the square pad which designates pin 1 of a square and the void is a round hole modeled as a negative loop associated with a circle. The center of the circle is located at the centroid of the square.

NODE: VIEW 14	TITLE: EXTERNAL SHAPE	NUMBER:

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE: 8 Jan 1988 PROJECT: Conceptual Schema for Layered Electrical Products REV: NOTES: 1 2 3 4 5 6 7 8 9 10	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT: S6, S7, S10, P6, P10, SS11



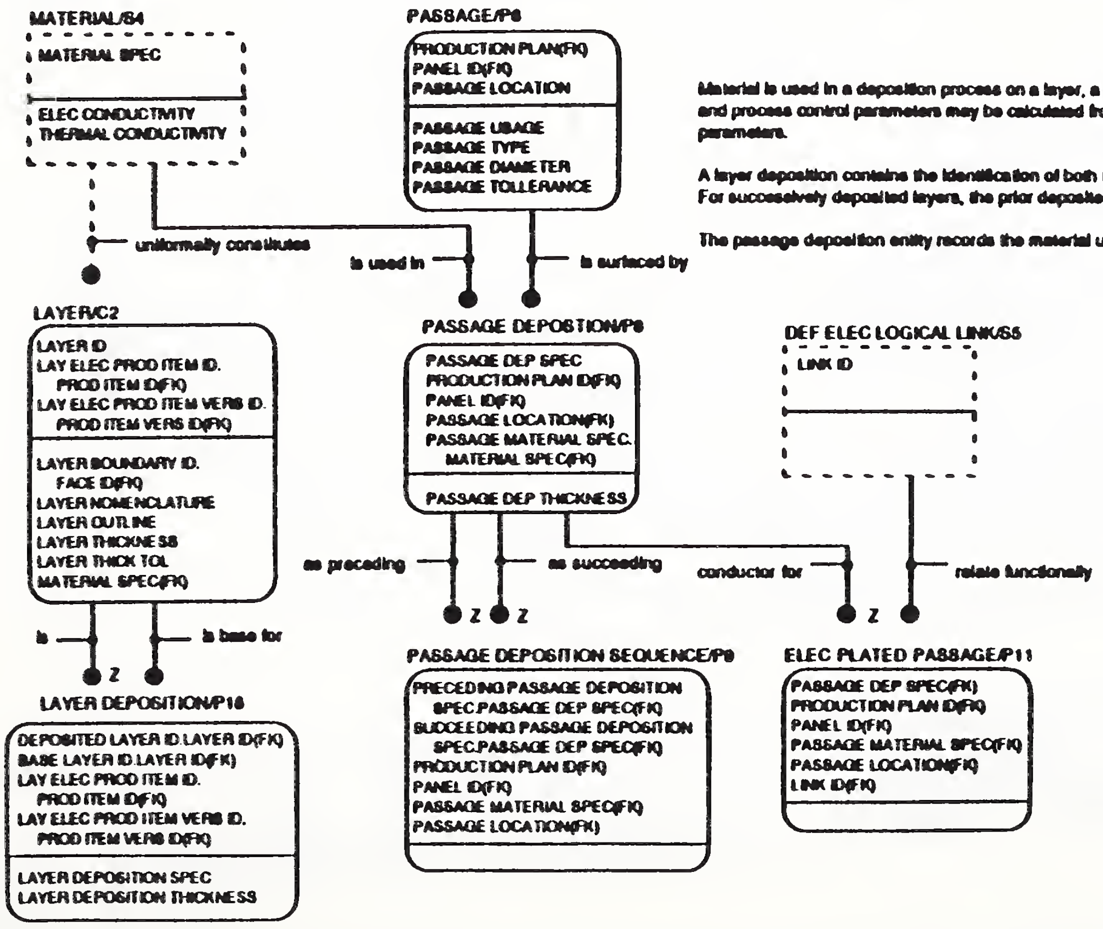
A component is a type of product item version which is identified as being part of the next assembly to the subject of this (Layered Electrical Product) model. As such, it may have Component Leads.

The Component Lead Vertex is the use of a Vertex for associating the lead with a place for topological connection. The component then is considered to participate in join topology (view 12) when the PWB is populated with components as an assembly.

A Passage may be related to a Component Lead with the Component Lead Passage entity.

NODE: VIEW 15	TITLE: COMPONENT	NUMBER:

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE	CONTEXT: S4, S5, P6, P8, P9, P11, P16, C2



Material is used in a deposition process on a layer, a passage, or both. Thickness is specified for each instance, and process control parameters may be calculated from area parameters and (external) process-specific parameters.

A layer deposition contains the identification of both the layer being deposited, and its base layer. For successively deposited layers, the prior deposited layer becomes the base layer.

The passage deposition entity records the material used for deposition on a passage.

NODE: VIEW 18	TITLE: DEPOSITION	NUMBER:

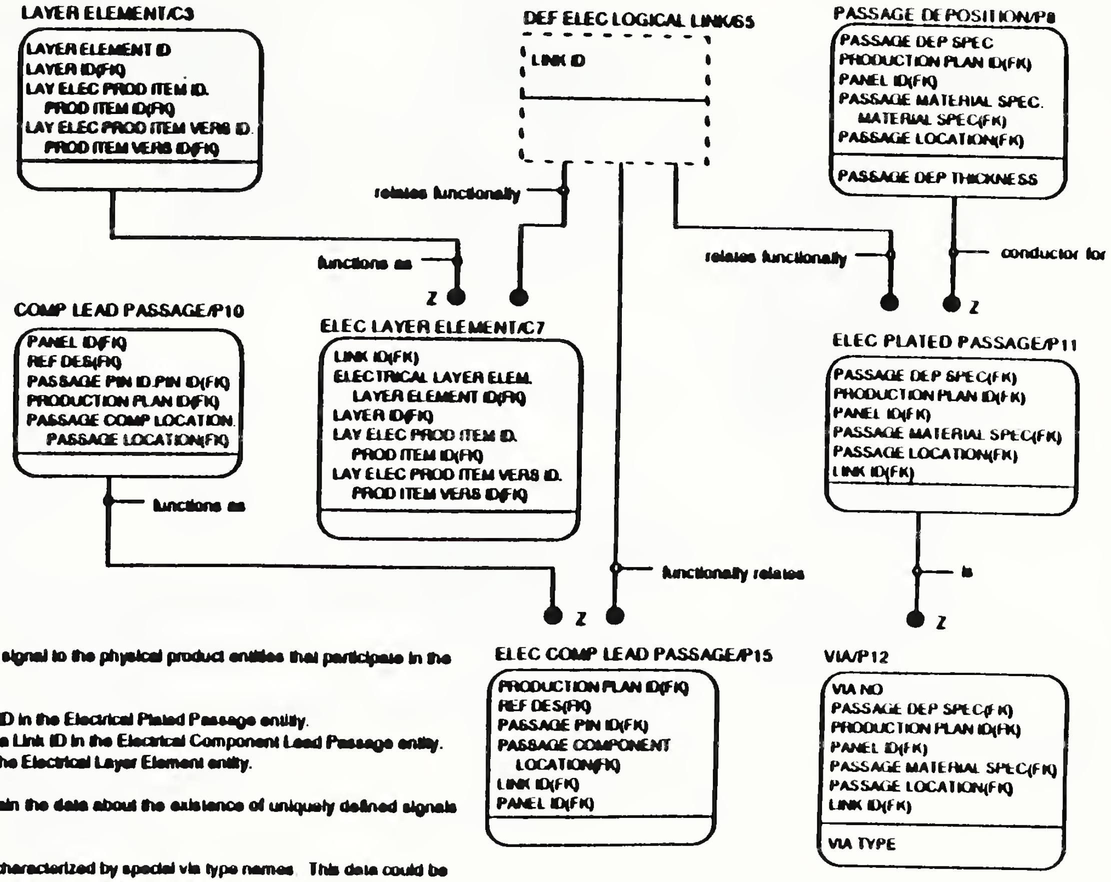
USED AT:

AUTHOR: IEEE/PDES Cal Poly Task Team Extension DATE: 8 Jan 1988
PROJECT: Conceptual Schema for Layered Electrical Products REV:
NOTES: 1 2 3 4 5 6 7 8 9 10

X WORKING
DRAFT
RECOMMENDED
PUBLICATION

READER DATE

CONTEXT: **
S5, P8, P18, P11,
P12, P15, C3, C7



This view brings together the entities which relate a signal to the physical product entities that participate in the signal connections.

- A Passage Deposition is associated with a Link ID in the Electrical Plated Passage entity.
- A Component Lead Passage is associated with a Link ID in the Electrical Component Lead Passage entity.
- A Layer Element is associated with a Link ID in the Electrical Layer Element entity.

Together, the above three association entities contain the data about the existence of uniquely defined signals as related to features of the product.

The via entity is a type of a plated passage which is characterized by special via type names. This data could be used for evaluating the design of the conductive trace routing in terms of eliminating vias by their type.

NODE: VIEW 17 TITLE: SIGNAL NUMBER:

USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE

CONTEXT:
C1, C2, P1, P2, P3, P4, P5, P8, S11,.

The Panel is a part defined by planning as a means of building the printed wiring board. Thus, it is brought into being by means of an order to produce that product. This information would be part of a manufacturing planning model.

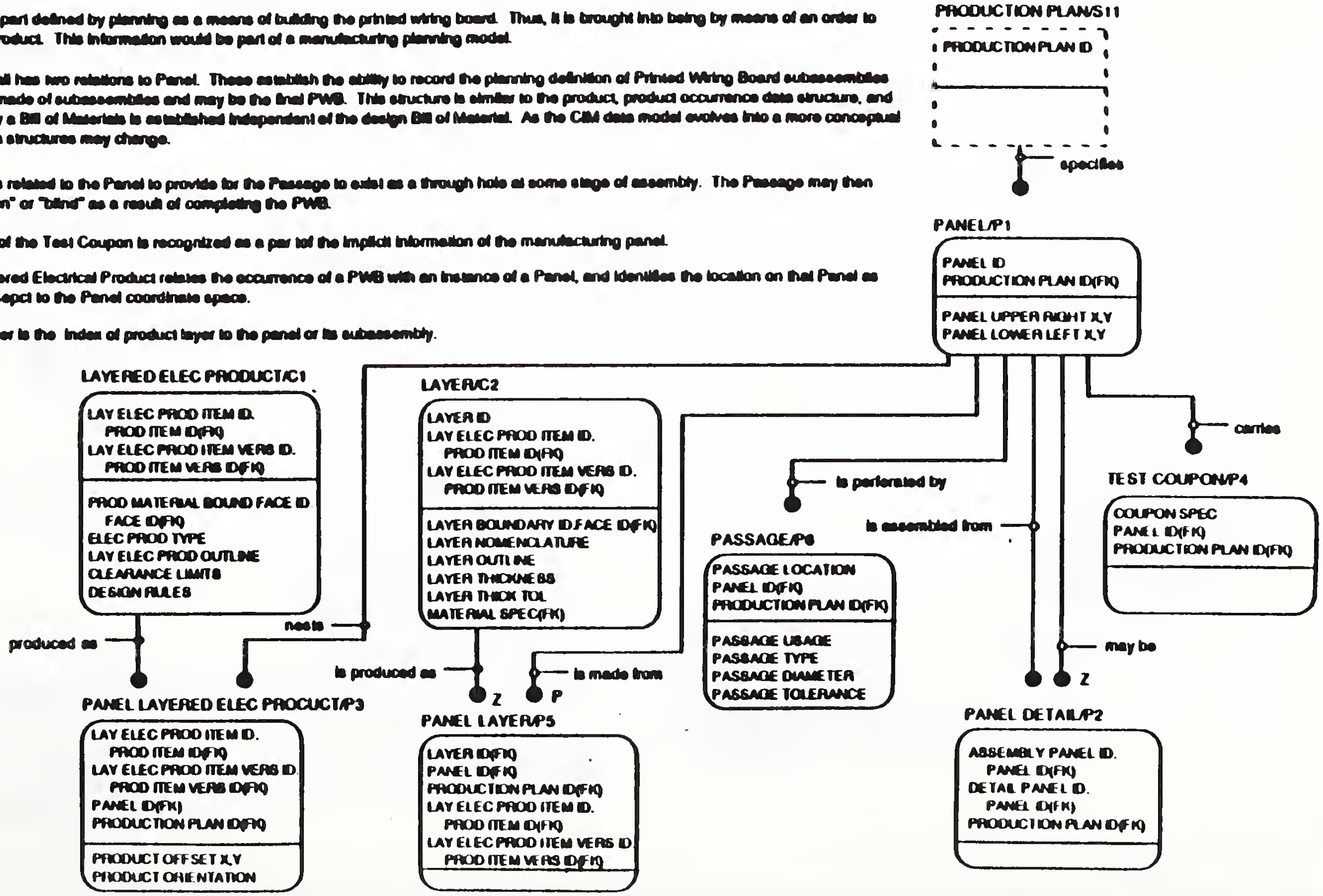
The Panel Detail has two relations to Panel. These establish the ability to record the planning definition of Printed Wiring Board subassemblies which may be made of subassemblies and may be the final PWB. This structure is similar to the product, product occurrence data structure, and reflects the way a Bill of Materials is established independent of the design Bill of Material. As the CIM data model evolves into a more conceptual form these data structures may change.

The Passage is related to the Panel to provide for the Passage to exist as a through hole at some stage of assembly. The Passage may then become "hidden" or "blind" as a result of completing the PWB.

The existence of the Test Coupon is recognized as a par tof the implicit information of the manufacturing panel.

The Panel Layered Electrical Product relates the occurrence of a PWB with an instance of a Panel, and identifies the location on that Panel as offsets with resepct to the Panel coordinate space.

The Panel Layer is the index of product layer to the panel or its subassembly.



NODE: VIEW 18
TITLE: PANEL
NUMBER:

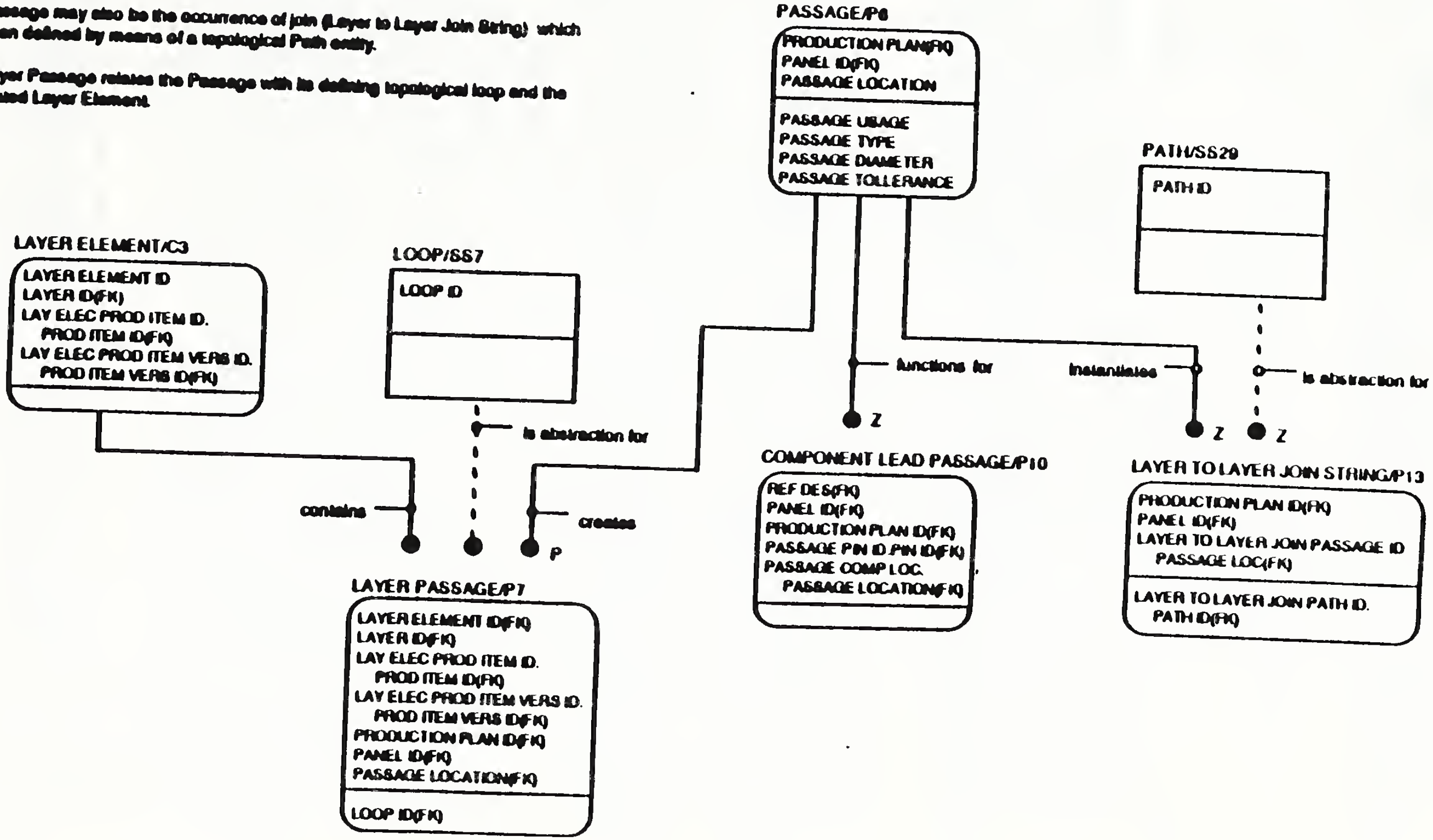
USED AT:	AUTHOR: IEEE/PDES Cal Poly Task Team Extension PROJECT: Conceptual Schema for Layered Electrical Products NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 8 Jan 1988 REV:	X WORKING DRAFT RECOMMENDED PUBLICATION	READER DATE

CONTEXT:
C3, P6, P7,
P10, P13, SS7, SS29

The Component Lead Passage relates occurrences of Passage instances with the Component Lead that is intended to be used in that passage.

The Passage may also be the occurrence of join (Layer to Layer Join String) which has been defined by means of a topological Path entity.

The Layer Passage relates the Passage with its defining topological loop and the associated Layer Element.



NODE: VIEW 10	TITLE: PASSAGE	NUMBER:

3.2 MODEL GLOSSARY

3.2.1 DESCRIPTION OF THE ENTITIES

Four distinct entity types can be found in the model. Each entity is identified by an entity indentification label which describes the entity type. Listed below are the entity identification labels. A brief description of each entity type follows.

ENTITY TYPE	ENTITY IDENTIFICATION LABEL
Core entities	C
Printed Wiring Board entities	P
Shadow entities	S
Shape/Size entities	SS

CORE ENTITIES: The Core entities can be found in the center of the global model. Collectively, these entities contain the conceptual description of what is generic to a Layered Electrical Product (LEP). They embody the second of the three levels of abstraction upon which the model has been built. The Core entities are less abstract than the Shape/Size portion of the model (to the left of the Core entities on the global model) and they are more abstract than the technology-specific solutions (to the right of the Core entities on the global model).

PRINTED WIRING BOARD ENTITIES: The Printed Wiring Board (PWB) entities which appear in this model are representative of a specific technology solution for electronic products that are defined in terms of layers. They have been included here as an example of one of several potential technology solutions which may be plugged into the right side of this model. (The PWB technology solution was chosen because there was some interest on the part of the Task Team's management in producing a PWB specific model and because there had already been some work done on PWB models by several of the particlpanting companies.) However, the team has done far more than develop a dedicated PWB model; it has developed a generic conceptual foundation on which several specific Layered Electrical Product (LEP) models can be developed. (Figure 3 illustrates the various technology solutions which are possible.)

SHAPE/SIZE ENTITIES: The shape/size model which was incorporated into this model was inspired by a topology model developed by Peter Wilson and Phil Kennicott at General Electrical Company in Schenectady, NY. Many of the shape/size entities which appear in this model are derivitives of concepts found in the Wilson-Kennicott model. Through the able interpretation (and patience) of Noel Christensen, these formal topological concepts were incorporated into the Cal Poly Task Team Extension model where they were used to describe the specific architecture of layered electrical products. It is somewhat misleading to refer to all of the entities in the Shape/Size partition of the model as "shape/size" entities because some of these entities, Surface/SS1, Face/SS2, Loop/SS7 and Edge/SS9, for example, have been directly taken from the discipline of Topology.

In addition to three entity types described above, the model includes "shadow" entities. (These are the boxes drawn in dashed lines on the global model.) These entities represent material which is related to but beyond the scope of the current modeling effort. They have been included here because they provide a place where other models may hook into our model.

3.2.2 ENTITIY DEFINITIONS

3.2.2.1 CORE ENTITIES

C1 LAYERED ELECTRICAL PRODUCT: A Physically Defined Product Item that has two or more Layers stacked in sequential order and who's primary functionality is electrical.

C2 LAYER: A stratum of one uniformly constituted Material the thickness of which may vary on one Layer. The vertical contours of the Layer may also vary.

C3 LAYER ELEMENT: A continuous topological region of topological dimension equal to two that is contained within one and only one topological layer.

C4 LAYER SEQUENCE: Ordered pairs of Layers which define overall ordering of all product Layers.

C5 ELEMENT SUBREGION: A subregion which is completely contained within a Layer Element. The Element Subregion may share zero or more boundaries of the Layer Element or of other Element Subregions.

C7 ELECTRICAL LAYER ELEMENT: A Layer Element that is associated with an electrical signal.

C8 ELEMENT JOIN SUBREGION: An Element Subregion that exists for the purpose of connecting Layer Elements. An Element Join Subregion can be connected to at most two Interlayer Joins.

C9 LAYER TO LAYER JOIN: A topological connection between exactly two Element Join Subregions that exist on two different Layers. Such a join may be effected by a bolt or conductive plating for example. More than one instance of this entity may be required at a given x,y location.

C10 COMPONENT LEAD JOIN: A connection between exactly one Element Join Subregion and a Component Lead.

C12 INFORMATION SUBREGION: A Logical Subregion that contains information that is represented graphically (I.e. Text, Icons, bar codes, ect.)

C14 LOGICAL SUBREGION: An artificial area which is superimposed anywhere on the Layer and does not nave to contain any material. A Logical Subregion may be partially or fully contained within another Logical Subregion. Special types of Logical Subregions include: Information Subregions and Restricted Subregions.

C15 AREA SHAPE: A Shape which has an arbitrary complex boundary that is defined explicitly by a Face.

C16 GRAPH SHAPE: A simple linear Shape that has a uniform width of which there are two types: Open Graph Shape and Closed Graph Shape.

C17 POINT SHAPE: A simple Shape that is located by a single Point, for example, a pad which is implicitly defined by, for example, a position on a photo plotter aperture wheel.

C18 INTRALAYER JOIN: A connection between exactly two Element Join Subregions that exist on the same Layer Element.

C22 ICON: A graphic contained within an Information Subregion (i.e. company logo, symbols, ect.)

C23 TEXT: Textual information represented graphically within an Information Subregion.

C28 LAYER ELEMENT SHAPE: The occurrence of a Shape associated with a Layer Element.

C29 RESTRICTED SUBREGION: A Logical Subregion which either fully or partially defines a keep-out boundary.

C30 EXPLICIT SHAPE: An explicit description defined by a Face, that can be applied to a Point Shape. It possess it's own coordinate system that is independent from the product. A typical example would be a library part resident in a CAD system.

C31 EXPLICIT ELEMENT POINT SHAPE: The occurrence of an Explicit Shape associated with a Point Shape.

C32 SHAPE: A description that can be applied to either Layer Elements or Logical Subregions thereby providing a boundary definition.

C33 LOGICAL SUBREGION SHAPE: The occurrence of a Shape associated with a Logical Subregion.

C34 CLOSED GRAPH SHAPE: A Graph Shape that is defined by a Loop.

C35 OPEN GRAPH SHAPE: A Graph Shape that is defined by a Path.

3.2.2.2 PRINTED WIRING BOARD ENTITIES

P1 PANEL: A rectangular or square base Material of pre-determined size intended for, or containing one or more Layered Electrical Products and, when required, one or more Test Coupons.

P2 PANEL DETAIL: The manufacturing identification of one or more Layers. The outside Layers may be processed to form a Layer Element, and Passages may be processed through the Panel Detail. The Passages may become blind or buried Vias when the Panel Detail is assembled to other Panel Details.

P3 PANEL LAYERED ELECTRICAL PRODUCT: An occurrence of a Layered Electrical Product in a manufacturing printed circuit or printed wiring board Panel. Note that one or more products are nested in a window area of the Panel.

P4 TEST COUPON: A portion of the quality conformance test circuitry used for a specific acceptance test or group of related tests.

P5 PANEL LAYER: The occurrence of a specific Layer as part of a Panel.

P6 PASSAGE: The implementation of a Void in a Panel Detail normal to the plane of its Layers. It is an implicit feature within the design topology and an explicit feature in the printed circuit or printed wiring board data. It may be drilled though the Panel Detail during some phase of detail processing depending on whether the element has been formed in order to be visually present or has been intended and formed to provide current conduction for plating purposes.

P7 LAYER PASSAGE: The occurrence of a Passage through a Layer.

P8 PASSAGE DEPOSITION: The occurrence of a specified Material deposited on a Passage.

P9 PASSAGE DEPOSITION SEQUENCE: An ordered pair of Passage Depositions.

P10 COMPONENT LEAD PASSAGE: The occurrence of a Passage intended for a Component Lead. The Passage may be plated.

P11 ELECTRICAL PLATED PASSAGE: The occurrence of an instance of an Electrical Logical Link on a Passage Deposition.

P12 VIA: An Electrical Plated Passage who's sole purpose is to electrically connect Layer Elements on two or more Layers. It can not be associated with a Component Lead.

P13 LAYER TO LAYER JOIN STRING: The occurrence of a Passage as a means of achieving one or more Layer to Layer Joins.

P15 ELECTRICAL COMPONENT LEAD PASSAGE: The occurrence of an Electrical Logical Link on a Component Lead Passage.

P16 LAYER DEPOSITION: A Layer who's Layer Elements are implemented as an additive process.

3.2.2.3 SHAPE/SIZE ENTITIES

SS1 SURFACE: Surface for Layered Electrical Product is assumed to be a plane. The plane is represented by a Point and a Direction.

SS2 FACE: An abstraction for a bounded portion of a Surface. A Face has a top, bottom, inside, outside, and a boundary. A Loop associated with the Face describes the boundary between the inside and the outside of the Face. Note that the Surface normal is used in coordinating the direction of the bounding Loop (see SS8). A Face defines a two dimensional topological region. In this region motion is restricted to forward, backward, and sideways but not up and down.

SS3 TRIMMING FACE: A Trimming Face is a Face that trims either another Face or a Surface. Trim means take a subset of the region of the trimmed object.

SS4 FACE STRUCTURE: A Face used by a Surface. It stands for the relationship between a Face and a Surface. It defines the relative direction between the Face and its associated Surface using the Face Structure Orientation attribute.

SS5 TRIMMED SURFACE: A Face that trims a Surface.

SS6 SUBFACE: A Face that trims another Face.

SS7 LOOP: An ordered collection of Edges with the following constraints: The Vertex of every Edge must be the same Vertex of exactly one other Edge. This means that an Edge Loop Structure defines a closed connection of Edges. It is used to bound Face/SS2. The positive direction of the Loop is arbitrarily assigned. However, once the positive direction of the Loop has been assigned, the directions of the underlying Edges must be coordinated to agree via Edge Loop Structure/SS10. This coordination marks the underlying Edge as either having the same or opposite direction of the Loop.

SS8 LOOP STRUCTURE: A Loop used by a Face. It stands for the relationship between a Loop and Face/SS2. It coordinates the direction of the Loop with the normal of the using Face through the convention of the right-hand rule. The right-hand rules states that if one is walking in the positive direction of the Loop, he is walking along the Loop on the top side of the Face and the inside of the Face is to the walker's left. If this is not true the Loop Structure coordinates the Loop direction by setting the Loop Structure Orientation attribute to negative.

SS9 EDGE: A connection between two Vertices (not necessarily distinct). An Edge does not describe physical details of the connection (it is an abstract connection). Usually an Edge has an interpretation in the physical world (a connecting conductor, part of the perimeter of the board etc.). One Vertex is said to be the start Vertex for the Edge and the other Vertex is said to be the stop Vertex for the Edge. A positive traversal of the Edge is said to be from the start Vertex to the stop Vertex. An Edge defines a topological region of dimension one. Movement within this region is restricted to forward or backward movement.

SS10 EDGE LOOP STRUCTURE: An Edge used by a Loop. It stands for the relationship between an Edge and a Loop. It coordinates the Direction of the Edge by indicating through the attribute Edge Loop Orientation whether the Direction of the Edge is the same or opposite of the using Loop. Positive means that when the Loop is being traversed in the positive direction, the particular underlying Edge is also being traversed in the positive direction. Negative means that when the Loop is being traversed in the positive direction, the particular underlying Edge is being traversed from stop to start Vertex.

SS11 VERTEX: An abstraction for something to be connected. A Vertex is said to be located, however a Vertex may initially be unlocated and the location associated later. Once an object has been abstracted by a Vertex, one may refer to the Vertex as if he were referring to the entire object in its full physical realization. A Vertex describes a region which is said to have topological dimension zero. This is a region which allows no degree of movement.

SS13 CURVE: A geometric one dimensional object. For Layered Electrical Products it is assumed to be either a Line or a Circle.

SS14 CURVE SEGMENT STRUCTURE: The trimming of a Curve by an Edge to create a Curve segment.

SS15 LINE: A Line is undefined. It is represented by a start Point and a Direction. A Line is unbounded.

SS16 CIRCLE: A Circle is undefined. It is represented by a center Point, a radius, and a Direction normal. For Layered Electrical Products the Direction is always positive z Direction. A Circle is always a complete Circle.

SS17 POINT: A Point is undefined. It is represented by an x and a y coordinate.

SS29 PATH: An ordered set of Edges. A Vertex of a participating Edge (except for the first and last Edges) must be the same as the Vertex of another participating Edge. The first and last Edge must have a single Vertex which is not the same Vertex of any other Edge within the Path. A Vertex may be the start/stop Vertex of at most two Edges within a Path. A Vertex within a Path may be the start Vertex of one Edge and the stop Vertex of another. A Vertex within a Path may be the stop Vertex for two Edges. A Vertex within a Path may be the start Vertex for two Edges. The positive direction of the Path is arbitrarily assigned. However, once assigned, the direction of the underlying Edges must be coordinated to agree with the Direction of the Path via Edge Path Structure. Positive means that when the Path is being traversed in the positive direction, the particular underlying Edge is being traversed from start to stop Vertex. Negative means that when the Path is being traversed in the positive direction, the particular underlying Edge is being traversed from stop to start Vertex.

SS30 EDGE PATH STRUCTURE: An Edge used by a Path. It stands for the relationship between an Edge and a Path. It coordinates the Direction of the Edge by indicating through the attribute Edge Path Orientation whether the Direction of the Edge is the same as or opposite to the Direction of the Path.

SS31 EDGE PATH STRUCTURE SEQUENCE: The sequence of an Edge in a Path. Given an Edge that participates in a Path, this entity refers to the preceding and succeeding Edge.

SS32 EDGE LOOP STRUCTURE SEQUENCE: The sequence of an Edge in a Loop. Given an Edge that participates in a Loop, this entity refers to the preceding and succeeding Edge.

SS33 DIRECTION: Direction is undefined. It is represented by its direction cosines i,j,k.

NOTE: The following shadow entities exist or will exist in another model. The definitions stated here are intended only to provide clarity as it relates to this model. For comprehensive definitions of these entities, refer to the appropriate models.

S1 PHYSICAL DEFINED PRODUCT ITEM: Information about an item which is intended to be, or is, included in the design definition of a product. The Physically Defined Product Item has related data which establishes its physical boundaries, or physical characteristics, or both. Everything from paint, bar stock, purchased screws and resistors to the highest defined assembly may appear as an instance of this entity. (see the PDES Planning Model).

S2 PHYSICAL DEFINED PRODUCT ITEM VERSION: Information about a variation of a Physically Defined Product Item. Every product has an original version and may have subsequent versions. One of the forms of identification of product versions is a change letter. (see PDES Planning Model)

S3 PHYSICALLY DEFINED CONSTITUENT ITEM OCCURRENCE: This is information about the basic relationship among items that establishes the structure of a product. Two different kinds of information can be drawn from this data: "bill of material" which is what items are required to make another item, and "where used" which is the information about all of the usages of a particular item. (see PDES Planning Model)

S4 MATERIAL: The tangible substance out of which a product is made. A Material is made of exactly one uniformly constituted chemical compound possessing consistent characteristics and properties. (see unknown future model)

S5 DEFINED ELECTRICAL LOGICAL LINK: Data about the logical electrical connectivity within a functional unit. When associated with Layer Elements, Passage Depositions, and Component Lead Passages in this model, it establishes two or more of these elements as being electrically in common. (see future Component/CCA model)

S6 COMPONENT: An individual part or combination of parts that performs a specific design function(s). It contains two or more Component Leads and can either be mounted on a Layered Electrical Product in a subsequent assembly process or be implemented as a Layer Element of the product. (see future Component/CCA model)

S7 COMPONENT LEAD: An element of a Component that serves as a mechanical and/or electrical connector to a Layered Electrical Product in an assembly. (see future Component/CCA model)

S8 INSPECTED PRODUCT UNIT: Information about a manufactured item which has been subjected to required examinations and tests and the results of those examinations and tests. (see PDES Planning Model)

S10 COMPONENT LEAD VERTEX: An abstraction for a Component Lead that is defined by a Vertex. (see future Component/CCA Model)

S11 PRODUCTION PLAN: The authorization and instructions for initiating the manufacture of an identified designed item-version. The plan is enterprise-specific, and defines the mechanisms needed for a production run of a defined size for a product. (see PDES Planning Model)

3.2.3 ATTRIBUTE DEFINITIONS

NOTE: The following set of attributes have been derived from the fully normalized version of the model (which is the version of the model after it has been run through JANUS). Note that some of the attributes appearing here may not appear on the global model (which is the C-size sheet enclosed with this report) due to a lack of space and a need to include only the essential elements on the global model. (This a not a flaw as many of the attributes listed below are not required to under the global model.) All of the attributes represented in this glossary are represented within the 19 model views found in Section 3.1 of this report.

CLEARANCE LIMITS: Design rules applied to the product as a whole that specify minimum clearance distances between layer elements (eg., anulus ring, overetch, conductor separation, registration, ect).

CONTINUITY: The existence or failure of a low resistance electrical path as specified between two points in a product unit.

COUPON SPEC: A specification (usually mil spec) describing the requirements for a portion of a panel used for destructive testing.

CURVE ID (CIRCLE ID): An identifier for a curve that is a circle.

CURVE ID (LINE ID): An identifier for a curve that is a line.

CURVE ID (TRIMMED CURVE ID): An identifier for a curve that has been trimmed by an edge.

CURVE ID: An identifier for a curve.

CURVE SEGMENT ORIENTATION: An indicator that specifies whether the curve and the edge have the same(= positive) or opposite(= negative) directions

CURVE TYPE: A categorization of curve. There are only two types of curves that we use in this model. They are lines and circles.

DESIGN RULES: Enterprise specific, self-imposed restrictions made necessary by product requirements and/or existing production capabilities (e.g., pad - pad, pad - trace, trace - trace, pattern - edge, grid values, supply volts default, signal volts max, signal current max, corner angle limit, conductor width min.).

DIRECTION I: A vector value along the x axis.

DIRECTION ID (CIRCLE NORMAL ID): An identifier for a direction that is used to define the normal plane of a circle.

DIRECTION ID: An identifier for a direction.

DIRECTION J: A vector along the y axis.

DIRECTION K: A vector along the z axis.

EDGE ID (COMPONENT LEAD JOIN ID): An identifier for an edge that is used as an abstraction for a component lead join.

EDGE ID (CURVE TRIMMING EDGE ID): An identifier for an edge that is used to trim a curve.

EDGE ID (INTRALAYER JOIN ID): An identifier for an edge that is used as an abstraction for an intralayer join.

EDGE ID (LAYER TO LAYER JOIN ID): An identifier for an edge that is used as an abstraction for a layer to layer join.

EDGE ID (LOOP EDGE ID): An identifier for an edge that is a constituent of a loop.

EDGE ID (PATH EDGE ID): An identifier for an edge that is a constituent of a path.

EDGE ID: An identifier for an edge.

EDGE LOOP ID (PRECEEDING EDGE LOOP ID): Not sure of the function of this in the edge loop structure sequence entity. May be redundant; see LOOP EDGE ID(PRECEEDING LOOP EDGE ID).

EDGE LOOP ID (SUCCEEDING EDGE LOOP ID): Not sure of the function of this in the edge loop structure sequence entity. May be redundant; see LOOP EDGE ID(SUCCEEDING LOOP EDGE ID).

EDGE LOOP ORIENTATION: An indicator that specifies whether the direction of an edge is the same (= positive) or opposite (= negative) when participating as an element of a given loop.

EDGE PATH ID (NEXT EDGE PATH ID): Not sure of the function of this in the edge path structure sequence entity. May be redundant; see PATH EDGE ID(NEXT PATH EDGE ID).

EDGE PATH ID (PRIOR EDGE PATH ID): Not sure of the function of this in the edge path structure sequence entity. May be redundant; see PATH EDGE ID(PRIOR PATH EDGE ID).

ELEC PROD TYPE: A categorization of layered electrical products. Examples include printed circuit boards, hybrid microassemblies, integrated circuits, flex harnesses, stripline boards, etc.

EXPLICIT ELEMENT POINT SHAPE ORIENTATION: Specifies the rotation of the coordinate space of an explicit shape relative to the coordinate space of the product.

EXPLICIT SHAPE ID (EXPLICIT ELEMENT SHAPE ID): An identifier for an explicit shape that describes a point shape.

EXPLICIT SHAPE ID: The identifier of a collection of geometry which defines an area. This identifier may be used as a library file name.

FACE ID (AREA SHAPE FACE ID): An identifier for a face that is used to define an area shape.

FACE ID (BOUNDED FACE ID): The identifier for a face as it participates with a loop(s) in a loop structure.

FACE ID (ELEMENT SUBREGION FACE ID): The identifier for a face that is used to define a subregion on a layer element.

FACE ID (EXPLICIT SHAPE NO): An identifier for a face that is used to describe an explicit shape.

FACE ID (LAYER BOUNDARY ID): An identifier for a face that is used to describe an imaginary boundary outside of which material for a given layer cannot exist.

FACE ID (MATERIAL BOUNDING FACE ID): An identifier for a face that is used to describe an imaginary boundary outside of which material for a given layered electrical product cannot exist.

FACE ID (TRIMMING FACE ID): An identifier for a face that is used to trim other faces or surfaces.

FACE ID (TRIMMING SUBFACE ID): An identifier for a face that has been trimmed by another face.

FACE ID: An identifier for a face.

FACE LOOP ORIENTATION: An indicator that specifies whether a face and a loop have the same (= positive) or opposite (= negative) directions.

FACE STRUCTURE ID: An identifier for a face structure.

FACE STRUCTURE ORIENTATION: An indicator that specifies whether a face and a shell have the same (= positive) or opposite (= negative) directions.

GRAPH SHAPE ASPECT: An indicator of whether the interior of the defining graph shape represents the existence (= positive) or lack (= negative) of layer material.

GRAPH SHAPE ID (CLOSED GRAPH SHAPE ID): An identifier for a graph shape that is a closed graph shape.

GRAPH SHAPE ID (OPEN GRAPH SHAPE ID): An identifier for a graph shape that is an open graph shape.

GRAPH SHAPE TYPE: A categorization of graph shapes of which there are two types: open graph shapes and closed graph shapes.

ICON ID: An identifier for an icon.

ICON TYPE: A descriptor that identifies different classifications of icons (ie logo, bar code, target, part polarity, component outline, etc.).

ISOLATION: The pass / fail of a specified high voltage test for electrical leakage between two non-connected metal areas of a product unit.

INFORMATION SUBREGION ID (ICON SUBREGION ID): The identifier of a topological area on the product which is intended to be used to contain special purpose graphics.

INFORMATION SUBREGION ID (TEXT SUBREGION ID): The identifier of a topological area on the product which is intended to be used to contain textual information.

LAYER DEPOSITION SPEC: A specification describing the requirements for a deposition process (ie. process, material, etc.).

LAYER DEPOSITION THICKNESS: The required material thickness for a given deposition layer.

LAYER ELEMENT ID (SHAPED LAYER ELEMENT ID): The identifier for a layer element that has a shape associated with it.

LAYER ELEMENT ID: An identifier for a layer element.

LAYER ELEMENT SUBREGION ID (VERTEX LAYER ELEMENT SUBREGION ID): An identifier for an element join subregion.

LAYER ELEMENT SUBREGION ID: An identifier for a layer element subregion.

LAYER ID (BASE LAYER ID): An identifier for a layer upon which a deposition is made.

LAYER ID (DEPOSITED LAYER ID): An identifier for a layer that is formed by deposition.

LAYER ID (NEXT LAYER ID): An identifier for a layer that follows the current layer.

LAYER ID (PRIOR LAYER ID): An identifier for layer that precedes the current layer.

LAYER ID (SUBREGION LAYER ID): An identifier for layer that contains one or more logical subregions.

LAYER ID: An identifier for a layer.

LAYER NOMENCLATURE: A descriptor define the type of layer (ie. conductor, stiffener, insulator, silkscreen, ground plane, power plane, soldermask, diffusion, etc.).

LAYER OUTLINE: A place holder: a set of curve values or a loop which defines the boundary of a layer.

LAYER THICKNESS TOLERANCE: The tolerance value applied to the material thickness of a layer.

LAYER THICKNESS: The material thickness of a layer.

LAYERED ELEC PROD ITEM ID: An identifier for a layered electrical product.

LAYERED ELEC PROD ITEM VERS ID: An identifier for a version of a particular layered electrical product.

LEP OUTLINE: A place holder: a set of curve values or a loop which defines the boundary of the product.

LINK ID: An identifier for a defined electrical logical link. Commonly referred to as a signal id.

LOGICAL SUBREGION ID (INFORMATION SUBREGION ID): An identifier for a logical subregion that is used as an information subregion.

LOGICAL SUBREGION ID (RESTRICTED SUBREGION ID): An identifier for a logical subregion that is used as a restricted subregion.

LOGICAL SUBREGION ID (SHAPED SUBREGION ID): An identifier for a logical subregion as it is described by one or more shapes.

LOGICAL SUBREGION ID: An identifier for a logical subregion.

LOOP EDGE ID (PRECEEDING LOOP EDGE ID): The first of a pair of edges which together define the order of edges which transverse a loop.

LOOP EDGE ID (SUCCEEDING LOOP EDGE ID): The second of a pair of edges which together define the order of edges which transverse a loop.

LOOP ID (EDGE LOOP ID): An identifier for a loop as it participates with an edge(s) in an edge loop structure.

LOOP ID (FACE BOUNDING LOOP ID): An identifier for a loop that is used to bound a face.

LOOP ID (GRAPH SHAPE BOUNDING LOOP ID): An identifier for a loop that is used to define a closed graph shape.

LOOP ID (SUBREGION LOOP ID): An identifier for a loop that defines the boundary for a logical subregion.

LOOP ID: An identifier for a loop.

MATERIAL SPEC (PASSAGE MATERIAL SPEC): A specification describing the requirements for material deposited in a passage.

MATERIAL SPEC: A specification describing the physical requirements for material used in the product.

MEASURED CLEARANCE: The value associated with a failure of a product unit due to clearance between conductors under the minimum specified.

PANEL ID (ASSY PANEL ID): An identifier for a panel that is also considered a detail.

PANEL ID (DETAIL PANEL ID): The numbers or characters used to identify separate subassemblies of a PWB for planning and manufacturing purposes. Sometimes called a "drawing dash number".

PANEL ID: A stock number or other identification for laminated, sized stock which is intended for use to manufacture a printed board.

PANEL LOWER LEFT XY: This attribute defines a corner of the panel, and can be used together with panel lower left x,y to determine the panel space and origin.

PANEL UPPER RIGHT XY: The panel is always assumed rectangular. This location defines one corner of the rectangle bounds.

PASSAGE DEPOSITION SPEC (PRECEDING PASSAGE DEPOSITION SPEC): The identification of the first of an adjacent pair of depositions which are successively made within a passage.

PASSAGE DEPOSITION SPEC (SUCCEEDING PASSAGE DEPOSITION SPEC): The identification of the second of an adjacent pair of depositions which are successively made within a passage.

PASSAGE DEPOSITION SPEC: The specification which defines the process to be used for depositing material within a passage.

PASSAGE DEPOSITION THICKNESS: The required material thickness for a given passage deposition.

PASSAGE DIAMETER: The required finished diameter of a passage.

PASSAGE LOCATION (LAYER TO LAYER JOIN PASSAGE ID): The location of a passage which is a layer to layer join passage.

PASSAGE LOCATION (PASSAGE COMP LOCATION): The identification of the component (by reference designator) which is intended to be associated with a passage instance.

PASSAGE LOCATION: Coordinate pair (x,y) that locates a passage with respect to a panel origin.

PASSAGE PIN ID: The identification of a pin of a component which is to be associated with an instance of a passage.

PASSAGE TOLERANCE: The acceptable limit of a finished passage from the specified diameter.

PASSAGE TYPE: A categorization of passage by functionality (ie, via passage, tooling passage, component lead passage, fixturing, registration, clearance, ect.)

PATH EDGE ID (NEXT PATH EDGE ID): The second of a pair of path edges which together define the order of edges participating in a path.

PATH EDGE ID (PRIOR PATH EDGE ID): The first of a pair of path edges which together define the order of edges participating in a path.

PATH EDGE ORIENTATION: An indicator that specifies whether the direction of an edge is the same (= positive) or opposite (= negative) when participating as an element of a given path.

PATH ID (EDGE PATH ID): An identifier for a path as it participates in an edge path structure.

PATH ID (LAYER TO LAYER JOIN PATH ID): An identifier for a path that is an abstraction for a layer to layer join.

PATH ID (OPEN GRAPH SHAPE PATH ID): An identifier for a path that is an abstraction for an open graph shape.

PATH ID: An identifier for a path.

PIN ID (PASSAGE PIN ID): The pin identification of a component which is intended to occupy the passage.

PIN ID: The identification of the instance of a lead or pin on a component. Usually numbered sequentially.

POINT ID (CIRCLE CENTER): An identifier for a point that is used to locate the center of a circle.

POINT ID: The unique identifier of a point.

POINT SHAPE ASPECT: An indicator of whether the interior of the defining point shape represents the existence (= positive) or lack (= negative) of layer material.

POINT SHAPE ID (EXPLICIT ELEMENT SHAPE POINT ID): An identifier for a point shape that is described by an explicit shape.

POINT X: The location of a point instance along the x axis.

POINT Y: The location of a point instance along the y axis.

PRODUCT ITEM ID (LAYERED ELEC PROD ITEM ID): The identification of a product which is classified as an electrical product.

PRODUCT ITEM VERSION ID (LAYERED ELEC PROD ITEM VERS ID): The identification of a version of a product which is classified as an electrical product.

PRODUCT LOCATION ID: An identifier for the location of a layered electrical product nested in a panel.

PRODUCT OFFSET XY: The location of the product origin with respect to the panel origin.

PRODUCT ORIENTATION: The rotation of the product as it is placed on a panel.

PRODUCTION PLAN ID: The identification of a piece of planning which will direct the manufacturing steps for a product.

RADIUS: The distance between the center of a circle and any given point on the circle.

REF DES: The identification of a place for a component within an assembly. Follows the guideline of ANSI Y32.16-1975 (eg., R1 = resistor 1, U7 = microcircuit 7, A10 = assembly 10).

SHAPE ID (AREA SHAPE ID): An identifier for a shape that is an area shape.

SHAPE ID (GRAPH SHAPE ID): An identifier for a shape that is a graph shape.

SHAPE ID (LAYER ELEMENT SHAPE ID): An Identifier for a shape that is a layer element shape.

SHAPE ID (POINT SHAPE ID): An identifier for a shape that is a point shape.

SHAPE ID (SUBREGION SHAPE ID): An identifier for a shape that is a subregion shape.

SHAPE ID: An identifier for a shape.

SHAPE TYPE: A categorization of shape. There are three classifications of shapes that we use in this model: point shape, graph shape, and area shape.

SUBFACE ORIENTATION: The rotation angle of a subface from its definition normal.

SURFACE DIRECTION: This should be a migrated attribute which defines the vector of an unbounded geometry surface.

SURFACE ID: An identifier for a surface.

SURFACE POINT: The relative origin of a surface.

SURFACE TYPE: A categorization of types when a surface is so designated.

TEXT ID: The identification of an instance of text.

TEXT STRING: The sequential string of characters that form the text content.

THERMAL CONDUCTIVITY: The rate at which heat is transmitted in the material.

TRIMMED FACE TYPE: A categorization of types when a trimmed face is so designated.

TRIMMED SURFACE ORIENTATION: The rotation angle of a trimmed surface from its definition normal.

TRIMMING FACE ID (FACE TRIMMING FACE ID): An identifier for a trimming face that trims another face.

TRIMMING FACE ID (SURFACE TRIMMING FACE ID): An identifier for a trimming face that trims a surface.

UNIT ID: An identifier for an inspected unit.

VERTEX ID (EDGE END VERTEX ID): The occurrence of an identified vertex which is the end of an edge instance.

VERTEX ID (EDGE START VERTEX ID): The occurrence of an identified vertex which is the start of an edge instance.

VERTEX ID (LAYER ELEMENT SUBREGION LOCATION ID): The occurrence of an identified vertex that is an abstraction of a region on a layer element that is intended for connection with another layer element or component lead.

VERTEX ID (SHAPE LOCATING VERTEX ID): The occurrence of an identified vertex that serves as an abstraction for a point shape.

VERTEX ID: The identifier for an instance of a vertex.

VIA NO: The integer identifier for an instance of a via.

VIA TYPE: A categorization of vias by their functionality and/or implementation (e.g., through product, hidden, blind, buried).

WIDTH: The assigned upon measured perpendicular to the direction of an open graph shape.

4. CONCLUSIONS AND RECOMMENDATIONS

4.1 QUESTION: What distinguished this modeling project?

4.1.1 Consistency of Modeling Staff: Basically the same group of people have been working since May. Very little time was required to bring new people up to speed. This consistency has improved communication within the group and has allowed us to learn from our mistakes as a team and to grow as a team. (This is beginning to sound like apple pie and motherhood!)

4.1.2 Riding out the Rough Times: Even with careful planning and scoping of the project it was difficult to find a core set of concepts upon which a stable model could be built. Having the patience to endure the unstable portion of the model development process is crucial to the survivability of the project. Modeling is an immature process. Each future modeling project enlarges the knowledge base.

4.1.3 Passion for Building Abstract Models: Because the scope of the project included a broad class of layered electrical product data such as printed wiring boards (PWBs), large scale integrated circuits (LSICs), and hybrid microelectronic circuits (HMCs), the team could not build a model that was biased toward a specific manufacturing technology. The original modeling approach, which focused on specific technology terms, built entity pools using the popular nomenclature found in specific products such as PWBs or hybrid designs). However, these distinct technical view points were tremendously diverse and did not encourage consensus or stability within the data model because, taken in context, it was nearly impossible to conclude that any one viewpoint was wrong. On the other hand, by developing an *abstract* basis on which to construct the model, we were able to remove some of the biases associated with a given technical viewpoint and to achieve a consensus (with a price). Although it was necessary to continually (mentally) map between the specific technical perspectives and the abstract description of these, this proved to be a kind of ongoing testing of the abstraction process by which we were able to deterimine if we were staying on course. Abstraction became an essential ingredient to achieving a stable model - the challenge was to build these abstractions while focusing on the modeling objectives.

4.1.4 Challenge to Achieve An Integrated Schema: The modeling team precisely defined the scope of the project but kept the model "open ended" knowing that they were modeling only a small piece of the complete electronic "picture." This open ended approach to modeling is distinguished by the integration of entities from beyond the stated domain of discourse such as product structure and production plan. (Entities falling outside the domain of discourse of this modeling project are drawn in dotted lines on the global model.)

4.1.5 Recognition of Integration Dimensions: The team recognized several dimensions of integration. These are: 1) integration across multiple technologies, 2) integration across disciplines of the life cycle, 3) integration across levels of meaning, and 4) integration of mulitple viewpoints.

4.1.6 Courage to Build a Layered Model: The team divided the model into three distinct layers: a fundamental layer to contain the abstract model; a generic layer to contain the general layered electrical product model; and a technology specific layer to contain important product-specific viewpoints. Each layer has a purpose and adds to the overall quality of the model. The layering was done without a formal mapping language by "hiding" mapping semantics in IDEF1X role names. For example the relationship, Face$_i$ 'is an abstraction for' elevates the Layer and Layered Electrical Product entities above the topological foundation on which the model is built. We understand that the proponents of non-redunancy will see these levels as redundant. We choose not to argue the point of redundancy. We found it useful.

4.1.7 Use of Model Qualification Techniques: The quality of a model can be measured by its ability to match the end users' mental image of the product he or she is designing. The end user (who is the knowledge expert) must be able to probe deeply into the model in order to make it his or her own. The basic tool for model qualification is a tool we refer to as Reference Path Analysis. (Note that Reference Path Analysis may be referred to as Query Analysis by some.) In Reference Path Analysis, the model's quality is measured by its ability to answer questions developed by the knowledge expert. The expert is encouraged to attempt to answer his questions based on traversing the model. A traversal is called a reference path, and it involves traveling from one entity using the relationship lines to another entity. (We use the metaphor of travelling from city to city with the roles being the roads between the cities.) Experts develop confidence in the model when they themselves do the reference path analysis. (However, we quickly find out what portions of the model the expert does not understand when he attempts reference path analysis.) This strengthens the bond between the modelers on the team and the expert (who may not be a seasoned modeler.)

4.1.8 Fortitude to Populate the Model: The model has been populated with data from a real production printed wiring board using approximately 15,000 data values. (This was not done by hand. Dzung Ha wrote a program which imported the data values received from the PDDI project into the database he was building in dBase III.) Model quality was improved through the population process. Even after thorough reading and theoretical evaluation of the model, discrepancies were discovered when populating the model. Some of these differences were found to be due the particular way that design data is stored in a CAD database. (See Appendix A4, issue # 18.)

4.1.9 Goal to Validate a Populated Model Using an Actual Database: We used a very conservative database technology (dBase III) to validate the populated model. In addition, dBase III was integrated with an inexpensive graphics package to allow us to graphically display query results using an IBM PC. The actual database used a compiled form of dBase III. This gave the modeling team a needed morale boost and supplied a degree of credibility to the uninitiated user.

4.1.10 The Future Value of the Conceptual Model: The integrated electrical product database simply does not exist today. We must do away with the kind of thinking that says that it is too difficult to achieve or that not enough people could ever agree on its structure. Our modeling group has proceeded on the assumption that integration is truely achievable. We must get out of the 'chicken and egg' loop and place an integrated model within the reach of the general public. We should not underestimate the public's ability to recognize quality and their capacity to innovatively use a quality model. But first a model must exist. We must recognize that the public perceives value in the ability to access information. We must make that (conceptual) information available.

4.1.11 Confidence in the PDES Approach: The PDES approach encourages the modeler to look for the common elements between similar domains of discourse and to build those common elements with standard (and fundamental) building blocks. The abstract layer of topology and geometry in our model comes close to resembling a standard building block. Our group did not wait for the Product Data Exchange Specification (PDES) organization to select a shape/size model. Any abstract shape/size model of reasonable quality that we could closely adhere to by constantly mapping the abstract shape/size definitions to the technology specific viewpoints was perceived to be better than no shape size model at all.

4.1.12 A Vision of a Standard CIM Model: We have a vision of bi-level standardization of an electrical model. This approach to standardization does not attempt to standardize the entire model. It may standardize only certain layers. One would think that the best chance for standardization would come at the more abstract layers. This allows for flexibility of the model by not standardizing at the level of 'user perception'. The level of user perception is the level of the model where the user begins to see terms that he is familar with, for example, plated-through hole, via, etc. However, there is a price for this kind of standardization. The user must have a helper (who understands the model) to assist him in mapping the standard partitions of the model to his specific user viewpoint. This cannot be done haphazardly. There must be a standard approach to the intepretation of the model which is supported by a *formal* language. Within this kind of environment, our model becomes a resource. This does not preclude standard viewpoints from developing. The emancipation of data from commercial CAD systems must occur before CIM can be achieved. This model is a step toward emancipating electrical product data from the way that it is represented today in most CAD systems.

4.1.13 A Modeling Team with Diverse Experience: This team had experience across several technologies (PWB, HMC, LSI) and working disciplines from electronic experts to seasoned modelers. Team members were willing to share their expertise.

4.1.14 High Quality Model Presentation: The quality of presentation of our model equals the quality of presentation found in product drawings. Large enough sheets were used to clearly read information about each entity. A CAD/CAM system was used to create this 'product' drawing.

4.1.15 The Team Organizational Structure: Initially, three baseline functional roles were established. These were a lead modeler, team administrator and team recorder. As time went on, the remaining participants assumed responsibilities which grew as the work of the team progressed. After a working model had been constructed, functions such as a database developer and administrator, query expert, and editor of the final report were assumed by individual team members. It is notable that our meetings of 8-12 people seemed to be a very productive number of participants.

4.1.16 Flexible Schedule: Our schedule of one-meeting-per-month was a realistic improvement over the previous team's schedule which met for three week sessions. Modeling is quite draining. Each of us benefited from the time at home that this schedule allowed to catch up and reflect between sessions. Rotating locations was a good way to involve local management. (Although at times, the hosting team member may have been called away or interrupted by his employer.)

4.2 QUESTION: What would we do differently?

4.2.1 Anticipate Technical Needs: About half-way through the project we discovered a glaring need for topology knowledge. We were lucky to have knowledge experts in this subject area nearby. These projects would do well to identify the areas of technology required and to staff accordingly.

4.2.2. Find a Neutral Computer Aided Manufacturing Installation: The value of available expertise on data needs for factory automation cannot be overstated. Most 'application experts' are in need of conceptual knowledge outside of the present means of doing business. A testbed, or place where automation is being conceptually explored would be an ideal meeting or lab site for this type of modeling acitivity.

4.2.3 IDEF1X Lessons Learned: Modeling in IDEF can be done in an 'only owned keys and attributes' mode. JANUS software is then used to migrate keys. However, the reviewers need to constantly keep in mind what information is ultimately contained in each entity. This requries a full model picture for reviews. In turn, the rule is that an entity must either be an intersection entity, or it must (only) exist to record an attribute that needs a home. Naming entities according to things that we are famillar with (because this produces easy, identifiable names) results in rather poor bulky models. This is perhaps a 'language learning curve' experience.

APPENDIX A1

TEAM CHARTER

March 18, 1988

CHARTER

CAL POLY TASK TEAM EXTENSION

An Industry Consortium (herin denoted "Team")
4 August 1987

Purpose

The team purpose is to define semantics and relationships required to effect computer integrated manufacturing (CIM) of electrical substrate-based products. As a consortium, Team members expect to bring the data model into their company computer architecture for acheiving CIM. The Team data model will be submitted to the Institute for Electrical and Electronics Engineers (IEEE) and the Product Data Exchange (PDES) organizations. Through review by these organizations, the consortium expects a closer data match between computer systems which are used to capture product data and the company CIM database.

Team membership is open to all who are interested in the stated purpose above. The Team will include as members any who have attended two meetings. The following companies and agencies who are recognized as members of this Charter are:

Allied Bendix
Dan Appleton, Co., Inc.
Eastman Kodak Company
General Dynamics
McDonnell-Douglas Astronautics Co.
National Bureau of Standards

Meetings

Meeting locations and agenda items shall be submitted to the IEEE/DASS (Design Automation Standards Subcommittee) and the PDES/EAC (Electrical Applications Committee) chairman by this Team for distribution to interested individuals as they see fit. These meetings of three to five days are scheduled approximately montly from May through December, 1987.

Cost

All Team members will bear their own cost of participation. Meetings may be hosted by members, and fees may be solicited by hosts for the purpose of offsetting meeting place costs. Member companies may provide printed material reproduction and computer time at no cost to the Team.

Liasion

In addition to the member companies and the IEEE and PDES standards organizations, the Team shall actively establish contacts with the following programs and encourage the involvement of the National Bureau of Standards (NBS) Automated Manufacturing Research Facility (AMRF), the NBS IEEE/CS Data Dictionary activities, the Integrated Information System Support (I^2S^2) program at Arizonia State University, the USAF Engineering Information Support (EIS), Computer Integrated Manufacturing (CIM), and Product Data Definition Interface (PDDI) programs. Additional programs, public and private, may also be included in the Team liasion.

Deliverables

The Team intends to produce a data model (not associated with any particular computer implementation) written in IDEF1X language as supported by the USAF Wright-Patterson Aeronautical Labs Computer Integrated Manufacturing program office. The model scope will be primarily the data about the physical electrical printed circuit board (PCB) as released from design, together with data which is added by planning and manufacturing. To whatever extent is practical, the model will also include hybrid microelectronic assemblies (HMAs) and integrated circuits (ICs). The model may also be expressed in the input language for JANUS, a computer program which supports the IDEF1X modeling language. The Team intends to prepare a final report and further intends to produce a relational database schema based on the model, together with data taken from a PCB test case. The latter is intended only to demonstrate a database implementation of the model and is not a recommended implementation. Working models will be published in IDEF1X 'kit' form and the final report will include the data model, glossary and issues addressed.

Duration

The Team will be dissolved upon producing the final report scheduled for March 18, 1988.

N288

APPENDIX A2

GENERAL RECORD OF DISCUSSION

March 18, 1988

GENERAL

This appendix is provided as documentation of the discussions which took place at the monthly working sessons. It is included so that readers may gain some sense of the arguments and analyses which occured in the process of creating this model. Keep in mind that the opinions represented here are not unamimous or even consensus views. In some cases, strong differences of opinion may exist.

Each monthly meeting consisted of one to four full days of modeling discussion. The remaining time was consumed by business issues and administrative activities. After the model was 'completed', sometime in early December, it was populated and a demonstration test case was developed.

1. ATOMIC DEFINITION OF FUNCTIONAL AND PHYSICAL UNITS: When attempting to capture form features, those features which are a member of an "unbreakable" Shape/Size Element Group are considered to be atomic physical units. Because design and manufacturing are working with a different set of form features, it is important to determine at what point a form feature is unbreakable. For example, design-wise a join may be considered unbreakable, however, from a manufacturing perspective, a join is something which does not have a part number, hence, from a manufacturing perspective, a join is not an atomic physical unit.

2. CONCEPTUAL DATA: Throughout the discussion, the requirements for conceptual data were frequently discussed. The following criteria were found to be characteristic of conceptual data.

    1. Conceptual data does not contain any specific usage ideas.
    2. Conceptual data does not contain any specific implementation ideas.
    3. Conceptual data does not contain data which is information computed from other data of the conceptual schema. For example, the value of a conceptual attribute may not be an algorithm.
    4. The meaning of an attribute must be in its definition not in the values it assumes.
    5. The meaning of an entity must be in its definition and not in the values of its key attribures.

    Further ideas about conceptual data were well expressed in the General Record of Discussions, Appendix 4, page 2 of the CAL POLY TASK TEAM FINAL REPORT, April 30, 1987. The following summary is taken from that report:

    > "A conceptual data model talks about data; it contains the *fundamental* data elements of an enterprise. Fundamental data elements are combined to create compounds. These compounds constitute the external views. The procedures for creating the compounds are unique recipes or instructions for manipulating elementary data. These procedures are unique entities in their own right. Are these recipes part of the conceptual data model? No.

> The conceptual data model does not tell one *how* to arrive at the data. Rather it depicts the relationships among the fundamental data items of the enterprise. For instance, a conceptual model does not contain the rules and procedures necessary to create a schematic (an external view of the raw data) although it does contain all of the data elements (and their interrelationships) necessary to construct this external presentation view of the functionality of the circuit. The rules of deriving external views are user-specific. The proper place for these rules might be the external schema."

3. ELECTRONIC COMPONENTS AND BOARD PATTERNS. Although electronic components are important in the printed circuit board assembly, they were avoided in the original Cal Poly Model because of their complex functional behavior. When examining the *physical* aspects of layered electrical products, however, one notes that many of the patterns found on a printed wiring board are the result of component architectures. For example, pad sizes are arranged to accomodate either component lead or body sizes. The idea of 'component projection' was developed to express the resultant effect that component architectures have on board patterns. Component projections appear in order to accomodate component leads or other factors. What we are interested in is that the component caused some impact on the phsycial description of the board, and by describing that impact, it becomes possible to capture all the information required to to model the assembled board.

4. ELECTRONIC PRODUCTS AS PHYSICAL PRODUCTS. At the first meeting, the team agreed to examine the physical aspects of electronic products. The focus on the physical nature of electrical products caused the functional idea of connectivity to be expressed by the physical idea of conductive pattern(s). Traces, pads, plated holes, and conductive areas were viewed as *physical* features resulting from some assembly process rather than as implementations of the functional network of which they are logically intended to connect.

5. LOOSE GEOMETRY. At the Kansas City meeting, it was noted that layer patterns contain elements that have no electrical significance. Calibration target is an example of this. These non-electrical layer pattern elements were dubbed 'loose geometry' and they are accounted for in the three types of logical subregions which have been delineated in the model, Restricted Subregion, Information Subregion and Logical Subregion Shape.

6. QUERY DEVELOPMENT: In order to help validate the model, each member of the team was encouraged to develop his or her own queries. Reference Path Analysis was then conducted to answer each query. The absence of any reference path or one that was too labyrinthine indicated places where the model was potentially weak. It is notable that most of the queries what were developed focused on the assembly, rather than the board itself. Several queries were found to extend beyond the scope of this project. The following methodology was employed as a guideline for query development. This methodology is based on four levels of query development: 1) natural Language, 2) conceptual queries, 3) sequential query language (SQL), and 4) a target system.

GUIDELINE TO QUERY DEVELOPMENT

I. NATUAL LANGUAGE: Understand the user's query
    a. Read the query and understand the user's terminology.
    b. Rewrite the query in terms that the modeler understands.

II. CONCEPTUAL QUERIES: Query Analysis
    a. Break the query into sub-queries by determining the operator and operands described in English.
    b. Determine if there are any overlapping sub-queries.
    c. Eliminate any sub-queries that are completely redundant.
    d. Note any queries that cannot be broken down any further.
    e. Further break into atomic segments these unique user queries.

III. STRUCTURED QUERY LANGUAGE: Map Semantics
    a. Reconcile the semantic components of the the user's query with the semantic components of a desired information model.
    b. Verify that we can find a logical access path for the query.
    c. Map the skeleton of the query to the skeleton of the information model.
    d. Itemize the queries that map to the information model.
    e. Query or compute.
    f. Logically optimize each query.

IV. TARGET SYSTEM: Perform the query using a populated database
    a. Map the logical query to a *target query language.*
    b. Physically optimize the query.
    c. Build a program for the non-query results.
    d. If desired, package the query in a 4th generation language or some other affable interface.
    e. Reconcile the results of these queries to the original requests.
    f. Map the results back to the user's terminology.

7. SHAPE/SIZE. Because the goal of the team is to capture physical information about layered electrical products, it is vital that the team develop an understanding of geometry. Topology was incorporated at the third session to express the conceptual nature of substrate-based electrical products. From there, a Shape/Size interface was developed that describes the specific physical nature of layered electrical products. Many hours were devoted to the discussion of Shape/Size entities and features. Referring again to the CAL POLY TASK TEAM FINAL REPORT, dated April 30, 1987, Appendix A3, issue #10, two key points were identified regarding shape/size:

> "The shape/size entity recognizes the existence of definitions of dimensions and tolerances for items which apply to more than one item. the existence of a multiple use shpae/size brings up the need tooidentify whose shape/size it is. That is, there is an owner of the definition who must be associated with it."

N288

APPENDIX A3

LIST OF PARTICIPANTS

March 18, 1988

THE CAL POLY TASK TEAM EXTENSION PARTICIPANTS

Name	Affiliation	Telephone
TEAM MEMBERSHIP:		
George Goldsmith	McDonnell-Douglas	(714)896-1712
Dzung C. Ha	McDonnell-Douglas	(714)896-2975
Kazuo Hori	McDonnell-Douglas	(714)896-1454
Mary Kathryn Kennedy	McDonnell-Douglas	(714)896-5264
Scott Madsen	General Dynamics	(714)868-4322
Robert J. Meagher	Eastman Kodak	(716)726-4512
Curtis Parks	General Dynamics	(714)868-4923
Joan Tyler	National Bureau of Standards	(301)975-6545
John Zimmerman	Allied Bendix	(816)997-2932
PARTICIPANTS:		
Richard Brooks	McDonnell-Douglas	(714)896-5161
Noel Christensen	Allied Bendix	(816)997-3984
Roger Gale	D. Appleton Co., Inc.	(213)546-7575
Yosef Haridim	Northrop	(213)332-8311
Darrell Kromarek	Boeing Electronics Co.	(206)657-8385
William Loye	Loye Associates	(612)644-8923
Paul Nelson	Hughes Aircraft Co.	(714)732-5008
Lawrence O'Connell	Sandia National Laboratories	(505)844-1061
Steven Piatz	UNISYS Corporation	(612)456-4438
ASSISTANCE:		
Dr. Richard Cockrum	Cal Poly ECE Dept.	(714)869-2510
Lt. Eric Gunther	Wright Aeronautical Labs	(513)255-6976
Jim Nell	Westinghouse	(301)993-5856
B. Neil Snodgrass	D. Appleton Co., Inc.	(817)571-8181

APPENDIX A4

LIST OF ISSUES

March 18, 1988

ISSUE #

1. 5/4/87

ISSUE: **ASSEMBLY**

DISCUSSION: One of the fundamental problems that this model needs to address is the issue of Assembly or how should the relationship between components which are in themselves assemblies be modeled?

STATUS: See Issue #5.

2. 7/13/87

ISSUE: **THE ARCHITECTURE OF LAYERED ELECTRICAL PRODUCTS**

DISCUSSION: At the third working session, several innovative ideas surfaced that would help the group represent the nature of substrate-based electrical products. At this time, it was noted that the group was in need of an architecture to portray both the sense of layered- and connected-ness which is characteristic of laminated assemblies. It was also noted that there is a need for a latice capability in order to map features which are related across assemblies.

STATUS: The Team has created a topological metaphor to express the nature of printed wiring assemblies. Entities such as Vertex, Edge, Loop, and Face have been incorporated into the model as a *conceptual* set of features with which the specific *physical* characteristics of layered electrical products including the notion of layering and some sort of connectivity which occurs between the layers can be expressed. In this model, topology functions as the conceptual architecture for layered electrical products.

3. 6/8/87

ISSUE: **BILL OF MATERIAL**

STATUS: The model adopts from the PDES PLANNING MODEL the notion that Physically Defined Product Items occur in versions. A Layered Electrical Product is one type of Physically Defined Product Item Version that may occur.

4. 7/13/87

ISSUE: **MAPPING TO THE DATABASE FROM A CONCEPTUAL SCHEMA**

DISCUSSION: There is a need for a formal way to understand the choice of a conceptual schema. Furthermore, today's computer systems may not be able to support a one to one contextual mapping from the conceptual schema to the database.

STATUS: Whenever possible, the group recognizes the need for the conceptual schema to become a component of the modeling language itself. However, resources do not exist that accomodate the concpetual schema within the modeling language itself. It seems that for the present, data exchange and information management will remain separate worlds.

NOTE: The demonstration test case which which was developed for this model can be somewhat misleading because although we have developed a conceptual model, in order to validate it (the model), we have actually gone one step further and populated this (conceptual) model by instantiating a database with data from an actual printed wring board assembly. The reader is cautioned to remember that this populated database was created soley to demonstrate the *practicality* of this information model. It was not intended as a recommended implementation of this material.

5. 7/13/87

ISSUE: **CONNECTIVITY**

DISCUSSION: How should the connectivity of layered assemblies be expressed?

RESOLUTION: There are mulitple layers and each layer has a part number, and there is an ordering to the layers so that the notion of 'make-from' versus the notion of 'assembly' is inherent in the model. A join has two stacks which is a sequence of 'make-froms'. *Connectivity* is the real issue.

6a. 6/8/87

ISSUE: **DOMAIN OF DISCOURSE AND MANUFACTURING CONCERNS**

DISCUSSION: At the initial meeting, there was a lot of discussion in order to define the 'Domain of Discourse' or what was to be the scope of this modeling effort. Although manufacturing processess and tools were seen to be clearly beyond the socpe of this modeling effort, it was at times difficult to keep manufacturing concerns from creeping into the model.

STATUS: There was an attempt to produce a *conceptually neutral* model that could be indepdently applied to any manufacturing process. Because features are the result of some manufacturing process, it is notable that there are very few features extant in the model. *Small History:* although via is now the only surviving feature in the model, the Team had originally started with a feature-laden entity pool. Pad, trace, land and plated through hole are among the entities that were prominent in the initial thinking of this modeling group. The team did not want to model manufacturing queries or tools because these are process-oriented concerns.

6b. 5/4/87

ISSUE: **DOMAIN OF DISCOURSE AND THE PHYSICAL INTERFACE**

DISCUSSION: At the June meeting, there was an attempt to further define our scope. One of these scoping decisions includes the decision to model either the physical interface or a part.

RESOLUTION: It was decided to capture the relationship between the physical world and the original Cal Poly model by delivering a good description of all layers, geometric relationships and holes with or without plating.

7. 7/13/87

ISSUE: **EXTENT**

DISCUSSION: This is a topological term found in the Peter Wilson model. It is controversial because the notion of extent does not explicity appear in conventional geometry or in manufacturing terminology. However, the notion of extent is an implied constraint which is inherent in the logical definition of many of the topologically-derived entities upon which our model has been constructed. For example, Layers, Panels, and Boards are all logically constrained by physical limitations. However, in formal topology, the definition of a Face (which is the basis for our Layer entity) has an unlimited planar extent.

RESOLUTION: Where logical constraints exist, due to the reality of the physical world, (such as what is the extent of a Layer), these have been documented in the definitions of the entities which can be found in the model glossary. Our specific notion of extent was designed to *implicitly* delimit the board.

8a. 5/4/87

ISSUE: **FEATURES: HIERARCHY**

DISCUSSION: There is some concern about features which violate the hierarchy of the modeling process. Note that many electrical components are organized in a networked fashion which is not in itself inherently heirarchical. We may be looking at the issue of managing complexity versus modeling complexity. An analogy would be the example where one considers a steering wheel versus the details of the steering mechanism (on an automobile for example.)

8b. 6/8/87

ISSUE: **FEATURES: TOPOLOGY**

DISCUSSION: Is the feature topology entity idependent from the shape/size topology entity?

RESOLUTION: It was thought that features are always interpretations of some shape/size. In the case of explicit features, some shape/size element groups are features. In the case of implicit features, it is possible to have unamed structures. In this model, the Team will discuss only those implicit features that are *named*.

8c. 6/8/87

ISSUE: **SPECIFIC FEATURES: JOIN, JOINT, THROUGH HOLE, TRACE**

DISCUSSION: The definition of a feature is a meaningful abstraction which has a name. The subject of generative planning develops a whole set of form features that are derived within the manufacturing process which are not necessarily the *same* features that the designer understands. Essentially, manufacturing and design are working with a *different* set of features.

Functionally, a join includes everything associated with it, but manufacturing-wise, a join is something without a part number. Within the enterprise, there is a need to emphasize the necessity to consider the 'as-built' part as distinct from the 'as-designed' part. Several concerns arise: for instance, is a plated through hole different from the traces it connects? (Note that a plated through hole occurs by deposition, however, with one unique distinction, that is, a plated-through hole is part of an interface, whereas, a trace does not participate in an interface).

STATUS: The team recognizes the need to separate features from processes because processes tend to evolve. The team realizes the advantages of conducting information modeling with features (because certain features are relatively stable despite the evolution of the manufacturing processes that have created them). However, the team also recognizes that features are not a fundamental element within the conceptual schema, and becasuse of this, has attempted to remove features from the model.

9. 6/9/87

ISSUE: **HYBRIDS**

DISCUSSION: Hybrids do not possess many of the features found in other layered electrical products. Should they be included in this model?

RESOLUTION: Physical features common to both hybrids and printed circuit boards have been included in the model. This model is suitable for hybrids to the (generic) extent that hybrids are layered electrical products. Physical issues which are unqiue to hybrids have not been modeled.

10. 7/13/87

ISSUE: **INTEGRATION OF 'GLUE ENTITIES'**

DISCUSSION: It has been recognized that those entities that link this model to other models belong neither in the functional, physical or conceptual worlds. This creates an integration dilemma, then, for at what point does the 'glue' between these three worlds begin to appear? At what point should integration occur?

STATUS: This issue needs to be resolved. Although the Team has integrated across several levels of meaning and technologies within the model, no integration has occured between this model and other models within PDES or any other organization. Once this external integration process is begun, it is not clear to what logical schema these 'glue' entities would belong. (Do they belong in the conceptual, internal or external schemas? Some of the 'glue' entities are the result of modeling in IDEF1X where intersection records have been created to signify the occurrence of two entities or give a home to foreign keys that must be migrated.) Other glue entities contain information which comes into existence as the result of the integration process. However, this information cannot be logically assigned to any one particular schema.

Integration Within the Model: Because our model contains four distinct entity 'super-types', the Core, Shadow, PWB and Shape/Size entities, a backbone of intersection records which 'zip' the application section of the model to the Shape/Size model and to certain points within the core model has been formed. These backbone entities effectively integrate the variety of entity types found in the model.

Integration External to this Model: Although places have been defined to provide for the integration of this model, (by allocating places for the Shadow entities within the model), at present, no integration of any material external to this project has occured. Models which may be linked to this model include the PDES Planning Level Model, the Cal Poly functional model, and possibly a (future) components model, when one becomes available.

11. 7/13/87

ISSUE: THE DESCRIPTION OF CONDUCTIVE AND NON-CONDUCTIVE INTERFACES

DISCUSSION: During the July session, it was noted that the group needed some daydreaming about what sort of topological features exist. In short, it became important to answer the question: what really is connecting? The problem was that in defining interface elements, these seemed to only work for the *conductive* elements.

STATUS: A Shadow entity, *Defined Electrical Logical Link*, has been added to the model. This entity is a hook into the Cal Poly functional model. It has been included here to provide a place for electrically significant layer passages. In this model, the idea of interface has been represented by the topological concepts of layer, layer element, and layer passage. Some layer elements and layer passages, those which have a relationship with electrical logical link, are electrical or conductive (layer) elements. This allows all interfaces, conductive and non- to be represented as a sequence of layers, layer elements and passages.

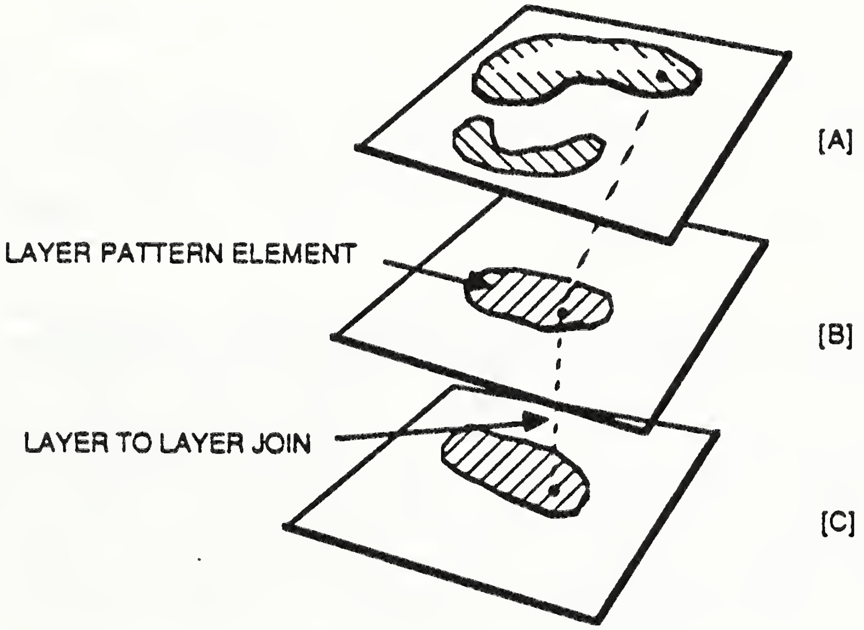
(See Issue #16 for an update on this.)

12. 6/10/87

ISSUE: THE DESCRIPTION OF PATH, JOIN AND JOINT

DISCUSSION: In the early versions of this model, a join could exist independently of a path or a joint. We are calling a joint the interfaces between terminals and lands, but IEEE includes all active components as well as the connections between them which would imply a continuous relationship. Is it a joint or a path that brings copper together on both sides of the board.?

RESOLUTION: Although a path can connect two lands, it should be contained within only one layer. This is different from the concept of a plated through hole which may connect several (serial) layers. Again, the idea of a join is a feature which has been removed from the model. What remains, are the concepts of layers, layer elements, layer patterns and layer passages. A join in this model may be abstractly represented by the presence of a vertex and the physical instance of a passage. Layer to Layer Join String, P13, allows several layers to be physically connected two at a time. This is illustrated below.



This is a simple illustration of a layered electrical product containing one inter-layer join. The inter-layer join shown here is made up of two layer-to-layer joins (one from layer [A] to layer [B] and one from layer [B] to layer [C].) Note that layer-to-layer joins are surface-sensitive, that is, a layer-to-layer join must be connected in such a way that the top of the bottom layer must be connected to the bottom of the top layer. (Or the reverse, the bottom of the top layer must be connected to the top of the bottom layer.) The filled areas in this illustration are layer pattern elements. Note that two layer pattern elements exist on layer [A]. A layer may contain several layer pattern elements.

Layer Topology

13a. 6/8/87

ISSUE: **MODELING APPROACH TO MULTI-LAYERED BOARDS**

DISCUSSION: At the second working session, it was noted that the group needed to establish some sort of time sequence of events so that data as it is contained in the physical description of a part library could be modeled. To do this required an interpretation of multi-layered boards.

STATUS: There was a motion to consider a multi-layered board to be merely an assembly of multi-layer parts. This would allow the third dimension to be acheived by the process of bonding and alignment.

(See Issue # 1 and 5.)

13b. 5/4/87

ISSUE: **MODELING APPROACH TO TOP-DOWN DESIGN**

DISCUSSION: There is a need for a methodology that is able to explain how the team is approaching top-down design.

STATUS: It was recognized that a model produced in IDEF1X needs to be linked to a hierarchy of models. By cleverly disguising a network of submodels (the Shape/Size model, LEP core model, an external application model) within this model, a limited hierarchical effect has been created. The Shape/Size model is the most abstract partition of the model, which supports a somewhat less abstract Core model, which in turn supports any number of external applications. However, the hierarchical effect which has been created within the model is local to this model, it does not address integration issues between this model and outside models. There is still a need to consider linking our model to the other models that have or will be developed within the PDES. Shadow entities have been suggested so that this model could be attached to a larger domain of discourse.

One of the drawbacks of the IDEF1X modeling language was the inability to express 'levels of abstractions' within the same model. There is still a need for a formal mechanism within the modeling language that will allow one to create an hierarchy of abstract discourses. By providing a link to the PDES Planning Model, the Team has, in effect, created a simple approach to Top-down design.

13c. 6/8/87

ISSUE: **MODELING LANGUAGE AND LEVELS OF ABSTRACTION**

DISCUSSION: The shortcomings of the relational model were brought out at many points in the working sessions. In particular, there was need for a modeling tool that allowed for levels of abstraction. Although it was possible to show several one-to-many relationships using IDEF1X, the real problem was to demonstrate abstractions *between* models. In particular, it became very important that things plug into the PDES Planning Level model.

STATUS: It was noted that the entity pools employed in this model do plug into high-level boxes on the PDES Planning Level model so that a rudimentary integration effect has been achieved.

14. 11/23/87 and 1/15/88

ISSUE: **PHYSICAL CONNECTION**

DISCUSSION: At both reviews of the model, it was noted that things which seem to be physically connected at the top is a connection of interest.

15. 1/15/88

ISSUE: **LOGICAL SUBREGION**

DISCUSSION: There is some question as to the proper parent of the Logical Subregion entity. As the model stands now, the Logical Subregion entity is owned by Shape, however, if Logical Subregions are wholly contained on Layers, which are an instantiation of a specific material bound for which the Face entity is an abstraction, should not the Face entity also be a parent of Logical Subregion?

16. 1/15/88

ISSUE: **RECONSTRUCTION OF THE MODEL TO ALLOW FOR PWB HOLES WHICH BEGIN AND END ON SPECIFIED LAYERS**

DISCUSSION: In attempting to populate a database with typical CAD system data, it was noted that the model as it is presently organized does not accomodate PWB holes, which begin on a specified layer and end on either the same or another specified layer. Conceptually, PWB holes are recognized by the model as negative loops contained within a Face. If there is extant manufacturing or planning data for a PWB hole available, the hole can *then* be recorded as a Passage which occurs within a level of assembly. However, at present, there is no place within our model to record information about PWB holes which do not pass through an entire level of assembly,

STATUS: It has been suggested that Layer become a non-identifying parent of Passage. This would allow for passages (holes) which begin and end on specific layers.

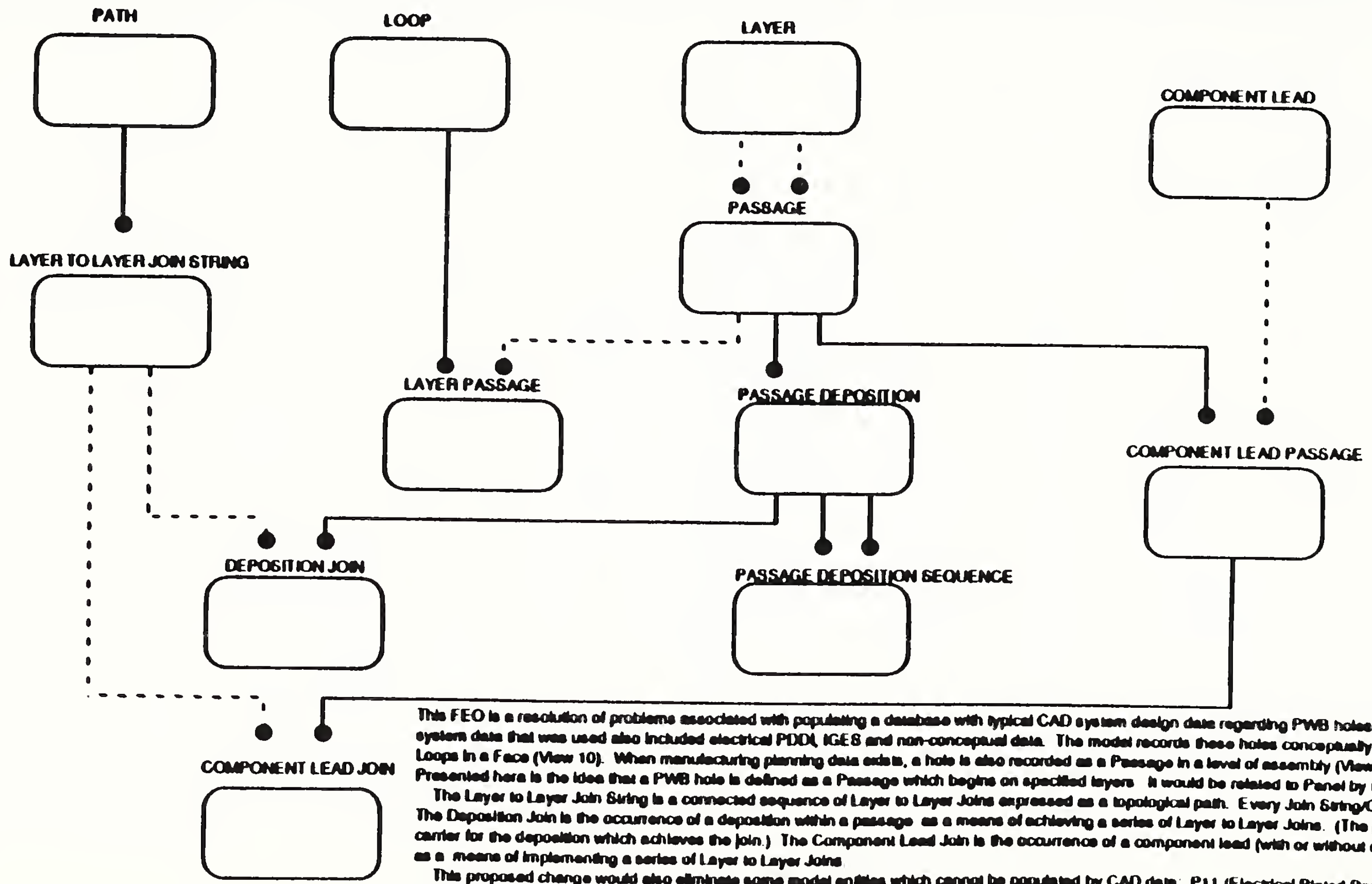
Further alterations to the model related to the support of this suggestion, include the addition of two intersection entities: Deposition Join and Component Lead Join. Deposition Join upholds the continuity of a Layer-to-Layer Join String in those passages where deposition is the only means of acheiving a Layer-to-Layer Join. In this case, the passage becomes the carrier for the deposition which achieves the join. Component Lead Join has been proposed as an intersection entity between Component Lead Passage and Layer-to-Layer Join String to accomodate Layer-to-Layer Join Strings which are implemented by the occurence of a component lead passage.

Certain entities could not be populated using typical CAD sytem data. These entities are: P11 (Electrical Plated Passage), P15 (Electrical Component Lead Passage which uses Electrical Layer Element/C7 and Component Lead Join/P10), C30 (Explicit Shape which uses Layer Passage/P7) and C31 (Explicit Element Point Shape). There is a move to remove the electrically significant passage entities because CAD system data makes no distinction between electrically or non-electrically significant passages. The Explicit Shape entities may be redundant because the intent of these entities can be accomodated by createing a non-idenifying relationship between Face/SS2 and Point Shape/C17 which allows for every Point Shape to be associated with a distinct Face.

The following FEO marked FX is a diagram of these proposed changes. As it stands now, the Team has proposed to submit this model to the PDES with this issue left unresolved.

USED AT:	**AUTHOR:** Cal Poly Task Team Extended **PROJECT:** Conceptual schema for layered electrical product data **NOTES:** 1 2 3 4 5 6 7 8 9 10	**DATE:** 4 Feb, 1988 **REV:**	X	**WORKING**	**READER**	**DATE**
				DRAFT		
				RECOMMENDED		
				PUBLICATION		

CONTEXT:



This FEO is a resolution of problems associated with populating a database with typical CAD system design data regarding PWB holes. The CAD system data that was used also included electrical PDDI, IGES and non-conceptual data. The model records these holes conceptually as Negative Loops in a Face (View 10). When manufacturing planning data exists, a hole is also recorded as a Passage in a level of assembly (View 18). Presented here is the idea that a PWB hole is defined as a Passage which begins on specified layers. It would be related to Panel by the entity P3.

The Layer to Layer Join String is a connected sequence of Layer to Layer Joins expressed as a topological path. Every Join String/C13 is a path. The Deposition Join is the occurrence of a deposition within a passage as a means of achieving a series of Layer to Layer Joins. (The passage is the carrier for the deposition which achieves the join.) The Component Lead Join is the occurrence of a component lead (with or without deposition) as a means of implementing a series of Layer to Layer Joins.

This proposed change would also eliminate some model entities which cannot be populated by CAD data: P11 (Electrical Plated Passage), P15 (Electrical Component Lead Passage which use C10 and C7), C30 (Explicit Shape which uses P7), and C31 (Explicit Element Point Shape). Add a non-identifying relationship between Face/SS2 and Point Shape/C17. (Thus every Point Shape has a distinct Face.)

NODE: FX	**TITLE:** PROPOSED CHANGE ON HOW TO HANDLE HOLES (VOIDS, PASSAGES) IN THE MODEL	**NUMBER:** A/ELEP/A

N288

APPENDIX A5

LIST OF SOURCE REFERENCE MATERIAL

March 18, 1988

Source #1: CAL POLY TASK TEAM FINAL REPORT, revised April 30, 1987

From: Roger Gale
Senior Consultant
D. Appleton Co., Inc.
13334 Park View Avenue, Suite 220
Manhattan Beach, CA 90266

Subject:
Functional Model for Electrical Products

Items Included:
1. Model Diagrams
2. Model Glossary
3. Model Business Rules
4. Narrative Explanation of the Model
5. Model Summary
6. Conclusions and Recommendations
7. Test Case Circuit and Data Instance Tables for the Model
8. Use of the Model to Evaluate IGES Connectivity
9. Issue List
10. General Record of Discussions
11. Activity Model for 'Design Electronic and Electrical Products'
12. List of Participants
13. List of Review Attendees
14. List of Source Reference Material
15. Lessons Learned on Team Modeling

Source #2: PDES PLANNING MODEL, revised 19 January, 1987

From: Roger Gale
Senior Consultant
D. Appleton Co., Inc.
13334 Park View Avenue, Suite 220
Manhattan Beach, CA 90266

Subject:
High-level Scoping Device for the PDES organization

Items Included:
1. Model Diagram

Source #3: PDES/IGES LOGICAL TOPOLOGY SCHEMA, May 1, 1987

From: Peter R Wilson and Philip R. Kennicott
General Electric Company
Corporate Research and Development
One River Road
Building 37, Room 549
Schenectady, NY 12345

Subject: Formal Definition of Topology

Items Included:
1. Additions and Proposed Changes to the Express Modeling Language
2. Definitions for Graph Traversal
2. Definitions for Topology

Source #4: PRODUCT DEFINITION FEATABASE FOR PRINTED WIRING BOARDS FINAL REPORT

From: Computer Aided Technology (McDonnell-Douglas Electronics Corporation)
December 1986

Subject: Product Definition for Printed Wiring Boards

Items Included:
1. Data Model
2. Model Glossary
3. PDDB-PWB Conceptual Database Entity Structure

Source #5: HUNTSVILLE MODEL

From: Lawrence O'Connell
Division 2812
CAE Integration
Sandia National Laboratories
5th and G Streets
Albuquerque, NM 87185

Subject: Product Structure Model for Printed Wiring Boards

Items Included:
1. Data Model

Source #6: SAN DIEGO MODEL

From: Lawrence O'Connell
Division 2812
CAE Integration
Sandia National Laboratories
5th and G Streets
Albuquerque, NM 87185

Subject: Product Structure Model for Printed Wiring Boards

Items Included:

1. Data Model

Source #7: THE PRODUCT DEFINITION DATA INTERFACE PROGRAM

From: McDonnell Aircraft Company
Box 516
St. Louis, MO 63166

Subject: Comprehensive program for the Computer Integrated Manufacturing of Electrical and Mechanical Parts

Items Included:

1. Data Model
2. Data for the Test Case Board

Source #8: ANSI/IPC-T-50C STANDARD, revision C, March 1985

From: The Institute for Interconnecting and Packaging Electronic Crcuits

Subject: Terms and Definitions for Interconnecting and Packaging Electronic Circuits

Items Included:

1. Glossary of definitions (some illustrations)

Source #8: IEEE Standard Dictionary of Electrical and Electronics Terms

From: ANSI/IEEE Std 100-1984 (third edition). Editor-in-Chief, Frank Jay.

Subject: Dictionary of Electrial and Electronic Terms

Items Included:

1. Dictionary of Terms

APPENDIX A6

STRUCTURAL MODELING LANGUAGE CODE

March 18, 1988

NOTE: The following code is a JANUS input file which describes the model. This code in this file matches the version of the model which is contained in the kit # A/ELEP/A, dated February 9, 1988.

```
VIEW PDES-ELEC-PWB
 GENERAL INFORMATION   SURFACE-TYPE
  AUTHOR PDES ELECTRICAL PWB TASK TEAM
  DESCRIPTION  LOGICAL SCHEMA FOR PRINTED WIRING BOARDS
   CREATION DATE  12/4/87
   REVISION DATE  12/4/87
  LEVEL  MSM

STRUCTURE

SHADOW ENTITY PHYSICALLY-DEFINED-PRODUCT-ITEM-VERSION
* ENTITY S2
 KEY
   PRODUCT-ITEM-ID
   PRODUCT-ITEM-VERSION-ID
 ENDKEY
ENDENTITY

SHADOW ENTITY INSPECTED-PROD-UNIT
* ENTITY S8
 KEY
  PHYSICALLY-DEFINED-PRODUCT-ITEM-VERSION "IS PRODUCED AS"
   UNIT-ID
 ENDKEY
  MEASURED-CLEARANCE
  CONTINUITY
  ISOLATION
ENDENTITY

ENTITY LAYERED-ELEC-PROD
* ENTITY C1
 KEY
  PHYSICALLY-DEFINED-PRODUCT-ITEM-VERSION Z "IS"
    ROLE LAYERED-ELEC-PROD-ITEM-ID FOR PRODUCT-ITEM-ID
    ROLE LAYERED-ELEC-PROD-ITEM-VERS-ID FOR PRODUCT-ITEM-VERSION-ID
 ENDKEY
  FACE Z NONULL "IS ABSTRACTION FOR MATERIAL BOUND FOR"
    ROLE MATERIAL-BOUNDING-FACE-ID FOR FACE-ID
  ELEC-PROD-TYPE
  LEP-OUTLINE
  CLEARANCE-LIMITS
  DESIGN-RULES
ENDENTITY
```

```
SHADOW ENTITY PRODUCTION-PLAN
* ENTITY S11
 KEY
   PRODUCTION-PLAN-ID
 ENDKEY
ENDENTITY

ENTITY PANEL
* ENTITY P1
 KEY
   PRODUCTION-PLAN "SPECIFIES"
    PANEL-ID
 ENDKEY
    PANEL-UPPER-RIGHT-XY
    PANEL-LOWER-LEFT-XY
ENDENTITY

ENTITY PANEL-DETAIL
* ENTITY P2
 KEY
   PANEL "IS ASSEMBLED FROM"
     ROLE ASSY-PANEL-ID FOR PANEL-ID
   PANEL Z "MAY BE"
     ROLE DETAIL-PANEL-ID FOR PANEL-D
 ENDKEY
ENDENTITY

ENTITY TEST-COUPON
* ENTITY P4
 KEY
   COUPON-SPEC
   PANEL "CARRIES"
 ENDKEY
ENDENTITY

ENTITY PASSAGE
* ENTITY P6
 KEY
   PANEL "IS PERFORATED BY"
   PASSAGE-LOCATION
 ENDKEY
   PASSAGE-TYPE
   PASSAGE-DIAMETER
   PASSAGE-TOLERANCE
   PASSAGE-USAGE
ENDENTITY
```

```
ENTITY PANEL-LAYERED-ELEC-PROD
* ENTITY P3
 KEY
   PANEL "NESTS"
   LAYERED-ELEC-PROD "IS PRODUCTED AS"
 ENDKEY
   PRODUCT-ORIENTATION
   PRODUCT-OFFSET-XY
ENDENTITY

ENTITY PANEL-LAYER
* ENTITY P5
 KEY
   PANEL P "IS MADE FROM"
   LAYER Z "IS PRODUCED AS"
 ENDKEY
ENDENTITY

ENTITY LAYER
* ENTITY C2
 KEY
   LAYERED-ELEC-PROD "HAS"
    LAYER-ID
 ENDKEY
   FACE Z NONULL "IS ABSTRACTION FOR MATERIAL BOUND FOR"
        ROLE LAYER-BOUNDARY-ID FOR FACE-ID
   MATERIAL NONULL "UNIFORMLY CONSTITUTES"
   LAYER-NOMENCLATURE
    LAYER-0UTLINE
   LAYER-THICKNESS
   LAYER-THICKNESS-TOLERANCE
ENDENTITY

ENTITY LAYER-DEPOSITION
* ENTITY P16
 KEY
   LAYER "IS BASE FOR"
     ROLE BASE-LAYER-ID FOR LAYER-ID
   LAYER Z "IS"
     ROLE DEPOSITED-LAYER-ID FOR LAYER-ID
 ENDKEY
   LAYER-DEPOSITION-SPEC
   LAYER-DPEOSITION-THICKNESS
ENDENTITY
```

```
ENTITY LAYER-SEQUENCE
* ENTITY C4
 KEY
   LAYER Z "SUCCEEDINGING"
     ROLE NEXT-LAYER-ID FOR LAYER-ID
   LAYER Z "PRECEEDING"
     ROLE PRIOR-LAYER-ID FOR LAYER-ID
 ENDKEY
ENDENTITY

ENTITY LAYER-ELEMENT
* ENTITY C3
 KEY
   LAYER P "HAS"
    LAYER-ELEMENT-ID
 ENDKEY
ENDENTITY

SHADOW ENTITY MATERIAL
* ENTITY S4
 KEY
    MATERIAL-SPEC
 ENDKEY
   ELEC-CONDUCTIVE
   THERMAL-CONDUCTIVE
ENDENTITY

SHADOW ENTITY DEF-ELEC-LOGICAL-LINK
* ENTITY S5
 KEY
     LINK-ID
 ENDKEY
ENDENTITY

ENTITY ELEC-LAYER-ELEMENT
*ENTITY C7
 KEY
   DEF-ELEC-LOGICAL-LINK "RELATES FUNCTIONALLY"
   LAYER-ELEMENT Z "FUNCTIONS AS"
 ENDKEY
ENDENTITY
```

```
ENTITY PASSAGE-DEPOSITION
* ENTITY P8
 KEY
   PASSAGE-DEPOSITION-SPEC
   MATERIAL "IS USED IN"
     ROLE PASSAGE-MATERIAL-SPEC FOR MATERIAL-SPEC
   PASSAGE "IS SURFACED BY"
 ENDKEY
   PASSAGE-DEPOSITION-THICKNESS
ENDENTITY

ENTITY ELEC-PLATED-PASSAGE
* ENTITY P11
 KEY
   PASSAGE-DEPOSITION Z "CONDUCTOR FOR"
   DEF-ELEC-LOGICAL-LINK "RELATES FUNCTIONALLY"
 ENDKEY
ENDENTITY

ENTITY COMP-LEAD-PASSAGE
* ENTITY P10
 KEY
   PASSAGE Z "FUNCTIONS FOR"
     ROLE PASSAGE-COMP-LOCATION FOR PASSAGE-LOCATION
   COMPONENT-LEAD Z "PLACED IN"
     ROLE PASSZGE-PIN-ID FOR PIN-ID
 ENDKEY
ENDENTITY

SHADOW ENTITY COMPONENT
* ENTITY S6
 KEY
   REF-DES
 ENDKEY
ENDENTITY

SHADOW ENTITY COMPONENT-LEAD
* ENTITY S7
 KEY
   COPONENT "HAS"
     PIN-ID
 ENDKEY
   LEAD-MAX-WIDTH
ENDENTITY
```

```
ENTITY ELEC-COMP-LEAD-PASSAGE
* ENTITY P15
 KEY
   COMP-LEAD-PASSAGE Z "FUNCTIONS AS"
   DEF-ELEC-LOGICAL-LINK "FUNCTIONALLY RELATES"
 ENDKEY
ENDENTITY

ENTITY VIA-PASSAGE
* ENTITY P12
 KEY
   ELEC-PLATED-PASSAGE Z "IS"
   VIA-NO
 ENDKEY
   VIA-TYPE
ENDENTITY

ENTITY PASSAGE-DEPOSITION-SEQUENCE
* ENTITY P9
 KEY
   PASSAGE-DEPOSITION Z "AS SUCCEEDING"
    ROLE SUCCEEDING-PASSAGE-DEPOSITION-SPEC FOR
        PASSAGE-DEPOSITION-SPEC
   PASSAGE-DEPOSITION Z "AS PRECEEDING"
    ROLE PRECEEDING-PASSAGE-DEPOSITION-SPEC FOR
        PASSAGE-DEPOSITION-SPEC
 ENDKEY
ENDENTITY

ENTITY VERTEX
* ENTITY SS11
 KEY
   VERTEX-ID
 ENDKEY
ENDENTITY

SHADOW ENTITY COMPONENT-LEAD-VERTEX
* ENTITY S10
 KEY
   COMPONENT-LEAD Z "IS ABSTRACTED AS"
 ENDKEY
   VERTEX Z NONULL "IS ABSTRACTION FOR"
ENDENTITY
```

```
ENTITY LOOP
* ENTITY SS7
 KEY
   LOOP-ID
 ENDKEY
ENDENTITY

ENTITY LAYER-PASSAGE
* ENTITY P7
 KEY
  PASSAGE P "CREATES"
  LAYER-ELEMENT "CONTAINS"
 ENDKEY
  LOOP NONULL "IS ABSTRACTION FOR"
ENDENTITY

ENTITY EDGE
* ENTITY SS9
 KEY
  VERTEX "IS START"
    ROLE EDGE-START-VERTEX-ID FOR VERTEX-ID
  VERTEX "IS END"
    ROLE EDGE-END-VERTEX-ID FOR VERTEX-ID
   EDGE-ID
 ENDKEY
ENDENTITY

ENTITY LAYER-TO-LAYER-JOIN
* ENTITY C9
 KEY
  EDGE Z "IS ABSTRACTION FOR"
    ROLE LAYER-TO-LAYER-JOIN-ID FOR EDGE-ID
 ENDKEY
ENDENTITY

ENTITY COMONENT-LEAD-JOIN
* ENTITY C10
 KEY
  EDGEZ "IS ABSTRACTION FOR"
    ROLE COMPONENT-LEAD-JOIN-ID FOR EDGE-ID
 ENDKEY
ENDENTITY
```

```
ENTITY INTRALAYER-JOIN
* ENTITY C18
 KEY
  EDGE Z "IS ABSTRACTION FOR"
   ROLE INTRALAYER-JOIN-ID FOR EDGE-ID
 ENDKEY
ENDENTITY

ENTITY EDGE-LOOP-STRUCTURE
* ENTITY SS10
 KEY
  LOOP P "PARTICIPATES IN"
   ROLE EDGE-LOOP-ID FOR LOOP-ID
  EDGE "PARTICIPATES IN"
  EDGE LOOP-EDGE-ID FOR EDGE-ID
 ENDKEY
  EDGE-LOOP-OREINTATION
ENDENTITY

ENTITY EDGE-LOOP-STRUCTURE-SEQUENCE
* ENTITY SS32
 KEY
  EDGE-LOOP-STRUCTURE Z "SUCCEEDING"
   ROLE SUCCEEDING-EDGE-LOOP-ID FOR EDGE-LOOP-ID
   ROLE SUCCEEDING-LOOP-EDGE-ID FOR LOOP-EDGE-ID
  EDGE-LOOP-STRUCTURE Z "PRECEEDING"
   ROLE PRECEEDING-EDGE-LOOP-ID FOR EDGE-LOOP-ID
   ROLE PRECEEDING-LOOP-EDGE-ID FOR LOOP-EDGE-ID
 ENDKEY
ENDENTITY

ENTITY LOGICAL-SUBREGION
* ENTITY C14
 KEY
  LAYER "CONTAINS"
   ROLE SUBREGION-LAYER-ID FOR LAYER-ID
  LOGICAL-SUBREGION-ID
 ENDKEY
  LOOP Z "IS ABSTRACTION FOR"
   ROLE SUBREGION-LOOP-ID FOR LOOP-ID
ENDENTITY
```

```
NTITY POINT
* ENTITY SS17
 KEY
  POINT ID
 ENDKEY
  VERTEX Z NONULL "IS LOCATED BY"
   POINT-X
   POINT-Y
ENDENTITY

ENTITY PATH
* ENTITY SS29
 KEY
   PATH-ID
 ENDKEY
ENDENTITY

ENTITY EDGE-PATH-STRUCTURE
* ENTITY SS30
 KEY
  PATH P "PARTICIPATES IN"
    ROLE EDGE-PATH-ID OR PATH-ID
  EDGE "PARTICIPATES IN"
    ROLE PATH-EDGE-ID FOR EDGE-ID
 ENDKEY
  EDGE-PATH-ORIENTATION
ENDENTITY

ENTITY EDGE-PATH-STRUCTURE-SEQUENCE
* ENTITY SS31
 KEY
  EDGE-PATH-STRUCTURE Z "SUCCEEDING"
    ROLE NEXT-EDGE-PATH-ID FOR EDGE-PATH-ID
    ROLE NEXT-PATH-EDGE-ID FOR PATH-EDGE-ID
  EDGE-PATH-STRUCTURE Z "PRECEEDING"
    ROLE PRIOR-EDGE-PATH-ID FOR EDGE-PATH-ID
    ROLE PRIOR PATH-EDGE-ID FOR PATH-EDGE-ID
 ENDKEY
ENDENTITY
```

```
ENTITY CURVE
* ENTITY SS13
 KEY
   CURVE-ID
 ENDKEY
 GENERALIZATION BY CURVE-TYPE OF
   CIRCLE
   LINE
 ENDGENERALIZAITON
ENDENTITY

ENTITY CURVE-SEGMENT-STRUCTURE
* ENTITY SS14
 KEY
   CURVE "IS TRIMMED BY"
     ROLE TRIMMED-CURVE-ID FOR CURVE-ID
   EDGE "TRIMS"
     ROLE CURVE-TRIMMING-EDGE-ID FOR EDGE-ID
 ENDKEY
   CRUVE-SEGMENT-ORIENTATION
ENDENTITY

ENTITY DIRECTION
* ENTITY
 KEY
   DIRECTION-ID
 ENDKEY
   DIRECTION-I
   DIRECTION-J
   DIRECTION-K
ENDENTITY

ENTITY CIRCLE
* ENTITY SS16
 CATEGORY BY CURVE-TYPE OF CURVE
     ROLE CIRCLE-ID FOR CURVE-ID
   POINT NONULL "IS CENTER"
     ROLE CIRCLE-CENTER FOR POINT-ID
   DIRECTION NONULL "NORMAL OF"
     ROLE CIRCLE-NORMAL-ID FOR DIRECTION-ID
   RADIUS
ENDENTITY .
```

```
ENTITY LINE
* ENTITY SS15
 CATEGORY BY CURVE-TYPE OF CURVE
     ROLE LINE-ID FOR CURVE-ID
     POINT NONULL "IS START OF"
     DIRECTION NONULL "ORIENTS"
ENDENTITY

ENTITY SHAPE
* ENTITY C32
 KEY
   SHAPE-ID
 ENDKEY
 GENERALIZATION BY SHAPE-TYPE OF
   POINT-SHAPE
   GRAPH-SHAPE
   AREA-SHAPE
 ENDGENERALIZATION
ENDENTITY

ENTITY RESTRICTED-SUBREGION
* ENTITY C29
 KEY
   LOGICAL-SUBREGION Z "IS"
     ROLE RESTRRICTED-SUBREGION-ID FOR LOGICAL-SUBREGION-ID
 ENDKEY
ENDENTITY

ENTITY INFORMATION-SUBREGION
* ENTITY C12
 KEY
   LOGICAL-SUBREGION Z 'FUNCTIONS AS"
     ROLE INFORMATION-SUBREGION-ID FOR LOGICAL-SUBREGION-ID
 ENDKEY
ENDENTITY

ENTITY ICON
* ENTITY C22
 KEY
   INFORMATION-SUBREGION "CONTAINS"
     ROLE ICON-SUBREGION-ID FOR INFORMATIN-SUBREGION-D
    ICON-ID
 ENDKEY
   ICON-TYPE
ENDENTITY
```

```
ENTITY TEXT
* ENTITY C23
 KEY
   INFORMATION-SUBREGION "CONTAINS"
     ROLE TEXT-SUBREGION-ID FOR INFORMATION-SUBREGION-ID
    TEXT-ID
 ENDKEY
   CHARACTER-STRING
   FONT
   HEIGHT
   ASPECT-RATIO
ENDENTITY

ENTITY LOGICAL-SUBREIGION-SHAPE
* ENTITY C33
 KEY
   SHAPE Z "IS"
     ROLE SUBREGION-SHAPE-ID FOR SHPAE-ID
   LOGICAL-SUBREGION P "IS DESCRIBED BY"
     ROLE SHAPED-SUBREGION-ID FOR LOGICAL-SUBREGION-ID
 ENDKEY
ENDENTITY

ENTITY LAYER-TO-LAYER-JOIN-STRING
* ENTITY P13
 KEY
   PASSAGE Z "INSTANTIATES"
     ROLE LAYER-TO-LAYER-JOIN-PASSAGE-ID FOR PASSAGE-LOCATION
 ENDKEY
   PATH NONULL "IS ABSTRACTION FOR"
      ROLE LAYER-TO-LAYER-JOIN-PATH-ID FOR PATH-ID
ENDENTITY

ENTITY LAYER-ELEMENT-SHAPE
* ENTITY C28
 KEY
   LAYER-ELEMENT P "IS DESCRIBED BY"
     ROLE S HAPED-LAYER-ELEMENT-ID FOR LAYER-ELEMENT-ID
   SHAPE Z "IS"
     ROLE LAYER-ELEMENT-SHAPE-ID FOR SHAPE-ID
 ENDKEY
ENDENTITY
```

```
ENTITY LAYER-ELEMENT-SUBREGION
* ENTITY C5
 KEY
   LAYER-ELEMENT "CONTAINS"
   LAYER-ELEMENT-SUBREGION-ID
 ENDKEY
ENDENTITY

ENTITY ELEMENT-JOIN-SUBREGION
* ENTITY C8
 KEY
   LAYER-ELEMENT-SUBREGION Z "IS"
     ROLE VERTEX-LAYER-ELEMENT-SUBREGION-ID FOR
        LAYER-ELEMENT-SUBREGION-ID
 ENDKEY
   VERTEX "IS ABSTRACTION FOR LOCATION OF"
     ROLE LAYER-ELEMENT-SUBREGION-LOCATION-ID FOR VERTEX-ID
ENDENTITY

ENTITY FACE
* ENTITY SS2
 KEY
   FACE-ID
 ENDKEY
ENDENTITY

ENTITY SURFACE
* ENTITY SS1
 KEY
   SURFACE-ID
 ENDKEY
   SURFACE-TYPE
   SURFACE-POINT
   SURFACE-DIRECTION
ENDENTITY

ENTITY TRIMMING-FACE
* ENTITY SS3
 KEY
   FACE Z "IS"
     ROLE TRIMMING-FACE-ID FOR FACE-ID
 ENDKEY
 GENERALIZATION BY TRIMMED-FACE-TYPE OF
   TRIMMED-SURFACE
  SUBFACE
 ENDGENERALIZATION
ENDENTITY
```

```
ENTITY TRIMMED-SURFACE
* ENTITY SS5
 CATEGORY BY TRIMMED-FACE-TYPE OF TRIMMING-FACE
    ROLE SURFACE-TRIMMING-FACE-ID FOR TRIMMING-FACE-ID
 SURFACE NONULL "IS TRIMMED TO"
 TRIMMED-SURFACE-ORIENTATION
ENDENTITY

ENTITY FACE-SRTRUCTURE
* ENTITY SS4
 KEY
   FACE "PARTICIPATES IN"
   FACE-STRUCTURE-D
 ENDKEY
   FACE-STRUCTURE-ORIENTATION
ENDENTITY

ENTITY LOOP-STRUCTURE
* ENTITY SS8
 KEY
   LOOP Z "PARTICIPATES IN"
     ROLE FACE-BOUNDING-LOOP-ID FOR LOOP-ID
   FACE P "IS BOUNDED BY"
     ROLE BOUNDED-FACE-ID FOR FACE-ID
 ENDKEY
   FACE-LOOP-ORIENTATION
ENDENTITY

ENTITY ELEMENT-SUBREGION-FACE
* ENTITY C25
 KEY
   LAYER-ELEMENT-SUBREGION Z "IS ABSTRACTED AS"
 ENDKEY
   FACE Z NONULL "IS ABSTRACTION FOR"
     ROLE ELEMENT-SUBREGION-FACE-ID FOR FACE-ID
ENDENTITY

ENTITY POINT-SHAPE
* ENTITY C17
 CATEGORY BY SHAPE-TYPE OF SHPAE
    ROLE POINT-SHAPE-ID FOR SHAPE-ID
 VERTEX Z NONULL "IS ABSTRACTION FOR"
    ROLE SHAPE-LOCATING-VERTEX-ID FOR VERTEX-ID
 ASPECT
ENDENTITY
```

```
          }

ENTITY AREA-SHAPE
* ENTITY C15
 CATEGORY BY SHAPE-TYPE OF SHAPE
    ROLE AREA-SHAPE-ID FOR SHAPE-ID
 FACE Z NONULL "IS ABSTRACTION FOR"
    ROLE AREA-SHAPE-FACE-ID FOR FACE-ID
ENDENTITY

ENTITY GRAPH-SHAPE
* ENTITY C16
 CATEGORY BY SHAPE-TYPE OF SHAPE
    ROLE GRAPH-SHAPE-ID FOR SHAPE-ID
 WIDTH
 ASPECT
 GENERALIZATON BY GRAPH-SHAPE-TYPE OF
    CLOSED-GRAPH-SHAPE
    OPEN-GRAPH-SHAPE
 ENDGENERALIZATION
ENDENTITY

ENTITY CLOSED-GRAPH-SHAPE
* ENTITY C34
 CATEGORY BY GRAPH-SHAPE-TYPE OF GRAPH-SHAPE
    ROLE CLOSED-GRAPH-SHAPE-ID FOR GRAPH-SHAPE-ID
 LOOP Z NONULL "IS ABSTRACTION FOR"
    ROLE GRAPH-SHAPE-BOUNDING-LOOP-ID FOR LOOP-ID
ENDENTITY

ENTITY OPEN-GRAPH-SHAPE
* ENTITY C35
 CATEGORY BY GRAPH-SHAPE-TYPE OF GRAPH-SHAPE
    ROLE OPEN-GRAPH-SHAPE-ID FOR GRAPH-SHAPE-ID
 PATH Z NONULL "IS ABSTRACTION FOR"
    ROLE OPEN-GRAPH-SHAPE-PATH-ID FOR PATH-ID
ENDENTITY

ENTITY EXPLICIT-SHAPE
* ENTITY C30
 KEY
   EXPLICIT-SHAPE-ID
 ENDKEY
  FACE Z NONULL "IS ABSTRACTION FOR"
    ROLE EXPLICIT-SHAPE-NO FOR FACE-ID
ENDENTITY
```

```
ENTITY EXPLICIT-ELEMENT-POINT-SHAPE
* ENTITY C31
 KEY
   POINT-SHAPE Z "IS DESCRIBED BY"
     ROLE EXPLICIT-ELEMENT-SHAPE-POINT-ID FOR POINT-SHAPE-ID
   EXPLICIT-SHAPE "IS DESCRIPTION OF"
     ROLE EXPLICIT-ELEMENT-SHAPE-ID FOR EXPLICIT-SHAPE-ID
 ENDKEY
   EXPLICIT-ELEMENT-POINT-SHAPE-OREINTATION
ENDENTITY

ENDSTRUCTURE

ENDVIEW
```

N28

APPENDIX A7

EXPRESS TRANSLATION OF THE MODEL

March 18, 1988

NOTE

This appendix contains a translation of the Cal Poly Task Team model into the EXPRESS modeling language. This translation was taken from the STEP/PC184-SC4-WG1 document authored by Peter R. Wilson and Philip Kennicott of General Electric Research Corporation, Schenectady, NY. It has been included here because the Cal Poly Task Team Extension wanted to make their model available in the same language of the specification of the STEP test draft. The translation contains three sections: geometry, topology and layered electrical products. It was produced from the C-size global model version by Peter Wilson and reviewed by John Zimmerman. It's presentation here does not necessarily signify complete agreement on the part of the Cal Poly Task Team Extension.

EXPRESS is an advanced data definition language (DDL). It was initiated under the USAF PDDI Program and was later used to support data definition for the PDES initiative within IGES. It's prinicple developers are Douglas Schenck (McDonnell-Dougals Corporation) and Bernd Wenzel (GEMAP MbH, Munich, FRG). EXPRESS is documented as ISO TC184/SC4/WG1:4:1.1 Document N177. It differs from IDEF1X in that EXPRESS describes objects, constraints, and operations for some universe of discourse. IDEF1X models only data, some constraints, and no operations.

The Cal Poly Task Team Extension is grateful to Peter Wilson for the opportunity to include this material.

Layered Electrical Product TIM

Peter R. Wilson

February 29, 1988

1 Introduction

This document provides a Topical Information Model for the Layered Electrical Product. The EXPRESS rendition was derived by Peter Wilson from the IDEF1X model prepared by the Cal Poly Task Team Extended and dated 8 January 1987. The first EXPRESS model was critiqued by John Zimmerman. This version incorporates the changes he suggested.

The EXPRESS *Integrated Product Information Model* (IPIM) has been used as an entity resource. The IDEF model contains definitions of entities — topology and geometry — that are in the IPIM. These entity definitions have not been included in this model; existing IPIM entities that are used in the IDEF model are listed later.

The EXPRESS model entities are not one-for-one with the IDEF entities, but hopefully this model is a faithful rendition of the information content of the IDEF. I have used "Editorial Comments" to explain some of the changes from IDEF to EXPRESS entities. It should be noted that not all the required constraints (WHERE clauses) have been included in the EXPRESS. These should be added later, after review of the model by the Cal Poly team.

1.1 UNDEFINED ENTITY

This entity is used in the model to capture the fact that something is undefined.

```
*)
ENTITY UNDEFINED;
END_ENTITY;
(*
```

1.1.1 Editorial Comments

1. This entity has been included in the EXPRESS IPIM (Washington Edition).

2 LAYERED ELECTRICAL PRODUCT MODEL

2.1 Scope

This model is intended to define the data about the as-designed electrical product which is expressible in terms of topological layers. Included are voids, boundaries and information contained on, or joining layers. Products which belong to this scope include integrated circuits, hybrid microelectronic assemblies, printed wiring (circuits and boards), flex harnesses and microwave stripline. In addition, entities are included which provide for the manufacture of printed wiring boards on a panel together with test coupons. The model as it relates to this scope is discussed in the model overview.

2.2 Purpose

The model was constructed by successive meetings between May and December 1987 of the Cal Poly team to achieve a common logical view of the data needed to drive factory automation (CIM) for the products mentioned above. In turn, the model is intended to become a contribution to the PDES/STEP integrated logical layer.

2.3 Model Overview

The CPTTX model is built on three levels of abstraction. Taking this approach gives the model a stable (and standard) foundation at the highest level of abstraction, generality at the middle level, and flexibility of expression at the highest level of abstraction This approach allows us to build the model assuring that each increment is of high quality. If the team cannot come to a consensus on any increment of the model, that increment is low in quality and the rest of the model will probably suffer until consensus is achieved. As a team we learned that we could not tackle the entire model at all levels at once and achieve a quality model. This approach gives structure to the overall model building process. First one asks "do we underestand the fundamental nature of the object we are modeling?" Once that fundamental structure is understood (independent of technology) in abstract terms, we hang more user friendly terminology on the abstractions, or leaves on the tree, to make the model assessable to persons not interested in the abstract foundations of the model. Once the model has been enriched with user friendly terminology, it is adapted to a specific type of technology, for example, LSI, HMA or PWB. In this phase of our effort, the model was adapted to PWB technology, however, hopefully, the model is general enough to be adapted to LSIs and HMAs as well.

Level 1: Fundamental Level (most abstract)

Level 2: Layered Electrical Product

Level 3: Technology Specific (least abstract)

2.3.1 LEVEL 1 — TOPOLOGY

A Layered electrical product (LEP) is abstractly represented as an assembly of topological faces. Each layer of a LEP is modeled as a distinct fact which bounds the material of that layer. This layer face contains sub faces which model the existence of only one material but the material can be broken into small pieces. The subfaces are in the same topological plane as the parent face or expressed in geometric terms, all material elements in a layer are co-planar. The sufaces themselves contain subfaces which are regions in which layer to layer joining occurs. Joining is a very general term for physical connectivity between layers. These connection regions are joined two at a time. Two regions joined together are referred to as *join* in this model. Joins are further abstracted as topological edges. The vertices of these edges are abstractions for the joined regions. The edge abstracts out all details of the process of joining while allowing us to state that a joining has occurred. In general the use of topology allows us to model the connectivity and boundedness of the LEP assembly without having to refer to geometry, specific technology, or manufacturing processes. The use of topology allowed the CPTTX group to come to an agreement on the basic assembly nature of an LEP without being involved in technology specific discussions.

2.3.2 LEVEL 2 — LAYERED ELECTRICAL PRODUCT

This level of the model describes the LEP in non topological terms but relates each term to entities in Level 1. Basically this level of the model states that a layered electrical product has one or more (sequential) layers — each layer containing one or more layer elements of the same material. Layer elements are further divided into basic shapes such as point shapes (pads), graph shapes (traces), and area shapes (ground planes). A layer element is created by pasting these basic shapes together. These basic shapes in turn relate directly to topological entities — a point shape corresponds to a simple face, a graph shape corresponds to a loop or a path, and an area shape corresponds to an aritrarily complex face. By doing this, we allow the layered electronic product expert to talk in terms of shapes instead of edges, vertices, faces, etc. In a sense these shapes are a type of feature which can be used as a handle far for more abstract notions. This is what features should do. This is the user friendly handle referred to earlier. Connections between layers are done with joins. Interlayer joins physically connect together element regions that are on different layers. Intralayer joins paste together the basic shapes on a layer. A component lead join physically ties a component lead to some region on a layer element (in our model the assumption is that the connection occurs on the element region closest to the body of the component). For convenience we also define logical regions on a layer for general purpose usage such as defining the region in which text is to be located. A logical region could define any region of interest, for example, a keep out region. It does not have to be bound material. In this layer of the model, layer elements are associated with electrical signals. However, the electrical signal is an entity on another model. We only show a reference to this other model by dotting the boxes that are referenced. The component is not a principal part of this model but it is referenced. Material is also referenced.

2.3.3 LEVEL 3 — PRINTED WIRING BOARD

This level of the model relates terminology used by a particular technology (e.g. for manufacturing planning) to the other two levels of the model. In our model that aspect is the particular way that joins are manufactured — as passages with deposition. Passages relate to voids (loops) in layer elements. Deposition (the plating inside the passage) is related to a set of joins that it implements. Vias are kinds of plated passages that join layer elements but do not accommodate component leads. Component leads themselves may provide the mechanism for joining. The important thing about this part of the model is that it describes objects that are closely related to the process of manufacturing (physically realizing) the concepts described in the other parts of the model. Level 3 can be "unplugged" and another model put in its place. This level of the model allows for flexibility for different viewpoints from different enterprises and different product types (HMA, LSI). It is not the intention that our level 3 is the ultimate answer, it is a typical perception of a PWB. Other level 3's could easily be wired in by relating the level 3 terms with their counterparts in level 1 or 2. In a sense, there is a kind of poor man's mapping or correspondence process which is occurring in our model. Also described in level 3 is the manufacturing object "panel" which illustrates the nesting relationship between an individual LEP and a panel with its panel layers, test coupons and panel detail. This represents a process planning view of a LEP. As an example, if someone else were to develop a level 3 application, they might want to refer to a silicon wafer instead of a panel, and a new set of child entities and level 2 relationships.

3 Level 1 Model

The Level 1 model consists of topological entities (vertex, edge, path, loop and face) and a restricted set of geometry entities (point, line, circle and (plane) surface). These entities do not appear in this document as their definitions are already available within the Integrated Product Information Model (IPIM). One additional topological entity — SUBFACE — is introduced by this Level.

3.1 SUBFACE

A SUBFACE defines a portion of a FACE.

```
*)
ENTITY SUBFACE SUBTYPE OF TOPOLOGY;
  OUTER_SUBFACE_BOUND : LOOP_LOGICAL_STRUCTURE;
  SUBFACE_BOUND       : SET [1 : #] OF LOOP_LOGICAL_STRUCTURE;
  TRIMMING            : SELECT_FACE_OR_SUBFACE;
WHERE
  OUTER_SUBFACE_BOUND IN SUBFACE_BOUND;
  EMBEDDED(SUBFACE, TRIMMING);
  CONSTRAINTS_FACE(SUBFACE);
END_ENTITY;
(*
```

```
*)
TYPE SELECT_FACE_OR_SUBFACE = SELECT (FACE, SUBFACE);
END_TYPE;
(*
```

ATTRIBUTE DEFINITIONS:

OUTER_SUBFACE_BOUND Outside boundary of SUBFACE. Optional.

SUBFACE_BOUND List of all the boundaries of the SUBFACE.

TRIMMING The FACE or SUBFACE that is being trimmed by the SUBFACE.

PROPOSITIONS:

1. The domain of the SUBFACE must be a subset of the domain of the FACE being trimmed.
2. All the constraints on FACE apply equally to SUBFACE

3.1.1 Editorial Comments

1. This entity has now been incorporated in the Washington Edition of the IPIM.
2. I have not used this entity consistently in the other parts of the model — instead I have used WHERE clauses to ensure that, for example, the FACE of a LAYER is effectively embedded within the FACE of a LAYERED ELECTRICAL PRODUCT. This is, I think satisfactory, as regards the Level 2 model.
3. Following from this, is it envisaged that a Level 1 model will ever "stand alone", i.e only the logical model is useful or is Level 2 the first useful model and there will always be both Level 1 and 2 models present?

   I incline to the thought that a Level 1 model is useful in its own right. If this is so, then there needs to be an entity that can serve as an "entry point" into the Level 1 model — perhaps something that just collects together the all the topological entities (vertex, edge, path and loop) that form the logical connectivity of the circuit.

   VERTEX mainly as an abstraction for an electrical component (resistor, capacitor, transistor etc).

   EDGE, PATH, LOOP as abstractions of connections between components. Vertices will be required in the definition of these entities, as usual.

4. As I see it, the Level 1 model is for the logical circuit layout and the Level 2 is for the physical layout of the circuit in a multi-layer environment. It is only at this level that FACES are needed in order to talk about physical areas of conductors etc. It should be possible, at a later stage to be able to unplug the Level 2 model from the Level 1 model and replace it with another physical environment model (e.g a power electrical distribution system).

4 Level 2 Model

This model is couched in terms more appropriate to the practising design engineer than to a topologist.

4.1 LAYERED ELECTRICAL PRODUCT

A LAYERED ELECTRICAL PRODUCT is a physically defined product that has two or more LAYERS stacked in sequential order and whose primary functionality is electrical.

```
*)
ENTITY LAYERED_ELECTRICAL_PRODUCT;
  MATERIAL_BOUND_ABSTRACTION    : FACE;
  PRODUCT_TYPE                  : LEP_PRODUCT_TYPE_ENUMERATION;
  OUTLINE                       : UNDEFINED;
  CLEARANCE_LIMITS              : UNDEFINED;
  DESIGN_RULES                  : UNDEFINED;
  LAYER_SEQUENCE                : ARRAY [1 : #] OF LAYER;
  JOINS                         : SET [0 : #] OF ASSEMBLY_JOIN;
```

```
WHERE
  PLANAR(MATERIAL_BOUND_ABSTRACTION);
  EMBEDDED(LAYER_SEQUENCE, MATERIAL_BOUND_ABSTRACTION);
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

MATERIAL_BOUND_ABSTRACTION The overall boundaries of the product. Material may only exist in this region.

LAYER_SEQUENCE The ordered sequence of the layers forming the LEP.

JOINS The set of joins between the layers of the product and between the product and external items.

PROPOSITIONS:

1. The geometry associated with the MATERIAL_BOUND FACE must be planar.
2. The individual layers must all be contained within the overall bound.

```
*)
TYPE LEP_PRODUCT_TYPE_ENUMERATION = ENUMERATION OF (RIGID,
                                                    FLEX_RIGID,
                                                    HYBRID,
                                                    INTEGRATED_CIRCUIT,
                                                    FLEX_HARNES);
END_TYPE;
(*
```

4.1.1 Editorial Comments

1. LAYER SEQUENCE/C4 has been made an attribute of this entity
2. The IDEF P cardinality on LAYER/C2 does not match the descriptive text. I think the text should be changed to include the possibility of defining a single-layered product, as in the EXPRESS and IDEF models.
3. I am not in favour of having TYPE information as part of an EXPRESS entity. Maybe SUPERTYPING/SUBTYPING should be used instead.
4. ASSEMBLY JOINs (i.e LAYER TO LAYER and COMPONENT LEAD JOINs) have been included in the information directly required for a LEP.

4.2 LAYER

A stratum of one uniformly constituted MATERIAL the thickness of which may vary on one LAYER.

A LAYER contains elements constituted of the same material. Layer elements may completely consume the region of the LAYER. A LAYER may contain voids.

```
*)
ENTITY LAYER;
  LAYER_BOUNDARY_ABSTRACTION : SELECT_FACE_OR_SUBFACE;
  LAYER_NOMENCLATURE         : LAYER_NOMENCLATURE_ENUMERATION;
  LAYER_OUTLINE              : UNDEFINED;
  LAYER_THICKNESS            : SIZE;
  MATERIAL_SPECIFICATION     : LEP_MATERIAL;
  LAYER_ELEMENTS             : SET [1 : #] OF LAYER_ELEMENT;
  LAYER_SUBREGIONS           : SET [0 : #] OF LOGICAL_SUBREGION;
  JOINS                      : SET [0 : #] OF INTRALAYER_JOIN;
WHERE
  EMBEDDED(LAYER_ELEMENTS, LAYER_BOUNDARY_ABSTRACTION);
  EMBEDDED(LAYER_SUBREGIONS, LAYER_BOUNDARY_ABSTRACTION);
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

LAYER_BOUNDARY_ABSTRACTION The domain of the LAYER. MATERIAL is limited to this domain.

LAYER_NOMENCLATURE The purpose of the LAYER.

LAYER_THICKNESS The thickness of the MATERIAL comprising the LAYER

MATERIAL_SPECIFICATION The specification of the material comprising the layer

LAYER_ELEMENTS The set of regions embedded in the LAYER

LAYER_SUBREGIONS The set of LOGICAL SUBREGIONS embedded in the LAYER

JOINS The set of joins within the LAYER.

```
*)
TYPE LAYER_NOMENCLATURE_ENUMERATION = ENUMERATION OF
                                       (GROUND_PLANE,
                                        POWER_PLANE,
                                        SILKSCREEN,
                                        SOLDER_MASK,
                                        DIFFUSION,
                                        ETC);
END_TYPE;
(*
```

PROPOSITIONS:

1. All LAYER ELEMENTS must be contained geometrically within the LAYER
2. All LOGICAL SUBREGIONS must be contained geometrically within the LAYER
3. LAYER ELEMENTs within a LAYER are disjoint (Need WHERE clause).

4.2.1 Editorial Comments

1. INTRALAYER JOINS have been made part of the information about a LAYER

4.3 LAYER ELEMENT

A continuous topological region of dimensionality 2 that is contained within one and only one LAYER.

A LAYER ELEMENT may have fully contained subregions. An electrical signal may be associated with a LAYER ELEMENT.

```
*)
ENTITY LAYER_ELEMENT;
  IS_DESCRIBED_BY : SET [1 : #] OF LEP_SHAPE;
  SUBREGIONS      : SET [0 : #] OF LAYER_ELEMENT_SUBREGION;
  SIGNAL          : OPTIONAL DEF_ELEC_LOG_LINK;
WHERE
  EMBEDDED(SUBREGIONS, LAYER_ELEMENT);
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

IS_DESCRIBED_BY The set of SHAPEs defining the LAYER ELEMENT region

SUBREGIONS The set of SUBREGIONS within the LAYER ELEMENT

SIGNAL An electrical signal that is associated with the LAYER ELEMENT

PROPOSITIONS:

1. The SUBREGIONS must be geometrically contained within the LAYER ELEMENT.

4.3.1 Editorial Comments

1. The IDEF entity ELECTRICAL LAYER ELEMENT/C7 is not in this model. Instead, the signal and element association is defined by the optional SIGNAL attribute.
2. SHAPE has been made an attribute of this entity and the IDEF entity LAYER ELEMENT SHAPE/C28 does not appear.

4.4 LAYER ELEMENT SUBREGION

A subregion which is completely contained within a LAYER ELEMENT. The ELEMENT SUBREGION may share zero or more boundaries of the LAYER ELEMENT or of other ELEMENT SUBREGIONS.

```
*)
ENTITY LAYER_ELEMENT_SUBREGION SUPERTYPE OF (ELEMENT_JOIN_SUBREGION);
  ABSTRACTED_AS : OPTIONAL SELECT_FACE_OR_SUBFACE;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

ABSTRACTED_AS The definition of the domain of the region

4.4.1 Editorial Comments

1. This has been made into a SUPERTYPE with one SUBTYPE. In the IDEF model this entity appears to have no "meaning" in the model — it is only the ELEMENT JOIN SUBREGION that has meaning. If I am wrong then no harm is done by this change.

4.5 ELEMENT JOIN SUBREGION

An ELEMENT SUBREGION that exists for the purpose of connecting LAYER ELEMENTS or a LAYER ELEMENT and a COMPONENT. An ELEMENT JOIN SUBREGION can be connected to at most two INTERLAYER JOINS.

```
*)
ENTITY ELEMENT_JOIN_SUBREGION SUBTYPE OF (LAYER_ELEMENT_SUBREGION);
  LOCATION    : VERTEX;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

LOCATION An abstraction of the location of the join region

4.6 LEP JOIN

A connection between two disjoint items.

The items that are connected are abstracted as vertices.

```
*)
ENTITY LEP_JOIN SUPERTYPE OF (ASSEMBLY_JOIN, INTRALAYER_JOIN);
  LOCATION_ABSTRACTION : EDGE;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

LOCATION_ABSTRACTION The location of the join

4.7 ASSEMBLY JOIN

```
*)
ENTITY ASSEMBLY_JOIN SUPERTYPE OF (LAYER_TO_LAYER_JOIN,
                                   COMPONENT_LEAD_JOIN)
                     SUBTYPE OF (LEP_JOIN);
END_ENTITY;
(*
```

4.7.1 Editorial Comments

1. LEP JOIN and ASSEMBLY JOIN have been included as SUPERTYPES for the purpose of being able, in other parts of the model, to talk about classes of JOIN types.

4.8 LAYER TO LAYER JOIN

A connection between exactly two ELEMENT JOIN SUBREGIONS that exist on two adjacent LAYERS.

```
*)
ENTITY LAYER_TO_LAYER_JOIN SUBTYPE OF (ASSEMBLY_JOIN);
WHERE
END_ENTITY;
(*
```

4.9 COMPONENT LEAD JOIN

A connection between exactly one ELEMENT JOIN SUBREGION and a COMPONENT LEAD.

```
*)
ENTITY COMPONENT_LEAD_JOIN SUBTYPE OF (ASSEMBLY_JOIN);
WHERE
END_ENTITY;
(*
```

4.10 INTRALAYER JOIN

A connection between exactly two ELEMENT JOIN SUBREGIONS that exist on the same LAYER ELEMENT.

```
*)
ENTITY INTRALAYER_JOIN SUBTYPE OF (LEP_JOIN);
WHERE
END_ENTITY;
(*
```

4.11 LOGICAL SUBREGION

An artificial area which is superimposed anywhere on a LAYER and does not have to contain any material. A LOGICAL SUBREGION may be partly or fully contained within another LOGICAL SUBREGION. Special types of LOGICAL SUBREGION include: INFORMATION SUBREGION and RESTRICTED SUBREGION.

```
*)
ENTITY LOGICAL_SUBREGION SUPERTYPE OF (RESTRICTED_SUBREGION,
                                        INFORMATION_SUBREGION);
  BOUNDARY         : LOOP;
  IS_DESCRIBED_BY  : SET [1 : #] OF LEP_SHAPE;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

BOUNDARY A loop defining the boundary of the region

IS_DESCRIBED_BY The shape of the region

4.11.1 Editorial Comments

1. Is there a tautolgy between the two attributes?
2. IDEF entity LOGICAL SUBREGION SHAPE/C33 does not occur in the EXPRESS model and LEP SHAPE becomes a direct attribute of this entity

4.12 RESTRICTED SUBREGION

A LOGICAL SUBREGION which either fully or partially defines a keep-out boundary.

```
*)
ENTITY RESTRICTED_SUBREGION SUBTYPE OF LOGICAL_SUBREGION;
WHERE
END_ENTITY
(*
```

4.13 INFORMATION SUBREGION

A LOGICAL SUBREGION that contains information that is represented graphically (i.e Text, Icons bar codes etc.)

```
*)
ENTITY INFORMATION_SUBREGION SUBTYPE OF LOGICAL_SUBREGION;
  ICONIC_INFORMATION  : SET [0 : #] OF ICON_PLACEMENT;
  TEXTUAL_INFORMATION : SET [0 : #] OF TEXT_PLACEMENT;
WHERE
  NOT (ICONIC_INFORMATION IS NULL AND TEXTUAL_INFORMATION IS NULL);
END_ENTITY
(*
```

ATTRIBUTE DEFINITIONS:

ICONIC_INFORMATION Icons to be placed in the region

TEXTUAL_INFORMATION Text to be placed in the region

PROPOSITIONS:

1. At least one of the attributes must not be NULL.

4.13.1 Editorial Comments

1. New entities have been defined for the placement of text and icons within the region

4.14 ICON PLACEMENT

A located instance of an ICON.

```
*)
ENTITY ICON_PLACEMENT;
  SYMBOL   : ICON;
  LOCATION : AXIS_PLACEMENT;
WHERE
END_ENTITY;
(*
```

4.15 ICON

A graphic (i.e company logo, symbols etc.)

```
*)
ENTITY ICON;
  TYPE_OF_ICON : UNDEFINED;
WHERE
END_ENTITY
(*
```

4.16 TEXT PLACEMENT

An instance of TEXT in a specific location.

```
*)
ENTITY TEXT_PLACEMENT;
  SYMBOL   : TEXT;
  LOCATION : AXIS_PLACEMENT;
WHERE
END_ENTITY;
(*
```

4.17 TEXT

Textual information represented graphically.

```
*)
ENTITY TEXT;
  CHARACTER_STRING : STRING;
  FONT             : UNDEFINED;
  HEIGHT           : REAL;
  ASPECT_RATIO     : REAL;
WHERE
END_ENTITY
(*
```

4.18 LEP SHAPE

A description that can be applied to either LAYER ELEMENTs or LOGICAL SUBREGIONS thereby providing a boundary definition.

If ASPECT is TRUE then the interior of the shape contains (LAYER) MATERIAL and if ASPECT is FALSE then the interior does not contain (LAYER) MATERIAL.

```
*)
ENTITY LEP_SHAPE SUPERTYPE OF (POINT_SHAPE, GRAPH_SHAPE, AREA_SHAPE);
  ASPECT : LOGICAL;
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

ASPECT Indication of MATERIAL or not

4.18.1 Editorial Comments

1. The IDEF entity SHAPE/C32 has been renamed LEP SHAPE to avoid conflict with other (potential) uses of the name.
2. The ASPECT attribute is applied to all shape types — in the IDEF model it did not apply to AREA SHAPE.

4.19 POINT SHAPE

A simple shape that is located by a single point.

```
*)
ENTITY POINT_SHAPE SUPERTYPE OF (EXPLICIT_POINT_SHAPE,
                                  IMPLICIT_POINT_SHAPE)
                   SUBTYPE OF (LEP_SHAPE);
  LOCATING_VERTEX : VERTEX;
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

LOCATING_VERTEX The (abstraction of the) location of the shape.

4.19.1 Editorial Comments

1. This has been made into a SUPERTYPE

4.20 IMPLICIT POINT SHAPE

A simple SHAPE that is located by a single POINT, for example, a pad which is implicitly defined by, for example, a position on a photo plotter apperature wheel.

```
*)
ENTITY IMPLICIT_POINT_SHAPE SUBTYPE OF (POINT_SHAPE);
  DEFINITION        : UNDEFINED;
  EXPLICIT_BOUNDARY : OPTIONAL FACE;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

DEFINITION The reference to the definition of the shape

EXPLICIT_BOUNDARY An explicit definition of the shape boundary.

4.20.1 Editorial Comments

1. The IDEF model contains nothing about how to represent/define a position on a photo aperture wheel.

4.21 EXPLICIT POINT SHAPE

An explicit description defined by a FACE, applied to a POINT SHAPE. It possess its own coordinate system that is independent from the product. A typical example would be a library part resident in a CAD system.

```
*)
ENTITY EXPLICIT_POINT_SHAPE SUBTYPE OF (POINT_SHAPE);
  BOUNDARY : FACE;
  LOCATION_AND_ORIENTATION : AXIS_PLACEMENT;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

BOUNDARY The domain of the shape

LOCATION_AND_ORIENTATION The local coordinate system of the shape

4.22 GRAPH SHAPE

A simple linear SHAPE that has a uniform width of which there are two types: OPEN GRAPH SHAPE and CLOSED GRAPH SHAPE.

Only the "center line" of the shape is represented; the "interior" of the shape is symmetrically disposed about the center line and has a total width of WIDTH.

```
ENTITY GRAPH_SHAPE SUPERTYPE OF (CLOSED_GRAPH_SHAPE, OPEN_GRAPH_SHAPE)
                   SUBTYPE OF LEP_SHAPE;
  WIDTH  : REAL;
WHERE
  WIDTH > 0;
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

WIDTH The width of the interior

PROPOSITIONS:

1. WIDTH must be greater than zero

4.23 CLOSED GRAPH SHAPE

A GRAPH SHAPE that is defined by a LOOP.

```
*)
ENTITY CLOSED_GRAPH_SHAPE SUBTYPE OF GRAPH_SHAPE;
  CENTER_LINE : LOOP;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

CENTER_LINE The center-line of the interior of the shape

4.24 OPEN GRAPH SHAPE

A GRAPH SHAPE that is defined by a PATH.

```
*)
ENTITY OPEN_GRAPH_SHAPE SUBTYPE OF GRAPH_SHAPE;
  CENTER_LINE : PATH;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

CENTER_LINE The center-line of the interior of the shape

4.25 AREA SHAPE

A SHAPE which has an arbitrary complex boundary that is defined explicitly by a FACE. The FACE may contain voids.

```
*)
ENTITY AREA_SHAPE SUBTYPE OF LEP_SHAPE;
  REGION : FACE;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

REGION The definition of the interior of the shape

5 Level 3 Model

This model is appropriate for the planning and manufacture of a Level 2 Model in the form of a Printed Wiring Board product. This model references the Level 2 Model.

5.1 PANEL

A rectangular or square base MATERIAL of pre-determined size intended for, or containing one or more LAYERED ELECTRICAL PRODUCTs and, when required, one or more TEST COUPONS.

```
*)
ENTITY PANEL;
  PANEL_UPPER_RIGHT  : CARTESIAN_TWO_COORDINATE;
  PANEL_LOWER_LEFT   : CARTESIAN_TWO_COORDINATE;
  COUPONS            : SET [0 : #] OF TEST_COUPON;
  PERFORATIONS       : SET [0 : #] OF LEP_PASSAGE;
  NESTS              : SET [0 : #] OF PANEL_LAYERED_ELECTRICAL_PRODUCT;
  COMPONENTS         : LIST [0 : #] OF PANEL_DETAIL;
  MADE_FROM          : SET [1 : #] OF LAYER;
WHERE
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

PANEL_UPPER_RIGHT The coordinates of the upper right end of the rectangle diagonal

PANEL_LOWER_LEFT The coordinates of the lower left end of the rectangle diagonal

COUPONS The set of TEST COUPONS on the panel

PERFORATIONS The set of perforations (through holes) in the panel

NESTS The set and location of LEPs on the panel

COMPONENTS The list of PANEL DETAILS (sub-assemblies) forming the panel

MADE_FROM The set of LAYERs forming the panel

5.1.1 Editorial Comments

1. The entity PANEL LAYER does not occur in this model. Instead, LAYER is referenced directly from PANEL.

5.2 PANEL DETAIL

The manufacturing identification of one or more LAYERS. The outside LAYERS may be processed to form a LAYER ELEMENT, and PASSAGES may be processed through the PANEL DETAIL. The PASSAGES may become blind or buried VIAS when the PANEL DETAIL is assembled to other PANEL DETAILs.

```
*)
ENTITY PANEL_DETAIL;
  DEFINITION : PANEL;
```

```
WHERE
END_ENTITY;
(*
```

5.3 PANEL LAYERED ELECTRICAL PRODUCT

An occurrence of a LAYERED ELECTRICAL PRODUCT in a manufacturing printed circuit or printed wiring board PANEL. Note that one or more products are nested in a window area of the PANEL.

```
*)
ENTITY PANEL_LAYERED_ELECTRICAL_PRODUCT;
  PRODUCT_OFFSET_AND_ORIENTATION : AXIS_PLACEMENT;
  PRODUCES                       : LAYERED_ELECTRICAL_PRODUCT;
WHERE
END_ENTITY;
(*
```

5.4 TEST COUPON

A portion of the quality conformance test circuitry used for a specific acceptance test or group of related tests.

```
*)
ENTITY TEST_COUPON;
  SPECIFICATION : UNDEFINED;
END_ENTITY;
(*
```

5.5 LEP PASSAGE

The implementation of a VOID in a PANEL DETAIL normal to the plane of its LAYERs. It is an implicit feature within the design topology and an explicit feature in the printed circuit or printed wiring board data. It may be drilled through the PANEL DETAIL during some phase of detail processing depending on whether the element has been formed in order to be visually present or has been intended and formed to provide current conduction for plating purposes.

```
*)
ENTITY LEP_PASSAGE SUPERTYPE OF (LAYER_PASSAGE, COMPONENT_LEAD_PASSAGE, VIA,
                                 LAYER_TO_LAYER_JOIN_STRING);
  LOCATION        : AXIS_PLACEMENT;
  USAGE           : OPTIONAL PASSAGE_USES_ENUMERATION;
  DIAMETER        : SIZE;
  IS_SURFACED_BY  : LIST [0 : #] OF DEPOSITION;
  SIGNAL          : OPTIONAL DEF_ELEC_LOG_LINK;
WHERE
```

```
END_ENTITY;
(*

*)
TYPE PASSAGE_USES_ENUMERATION = ENUMERATION OF (FIXTURING,
                                                REGISTRATION,
                                                MOUNTING,
                                                CLEARANCE);
END_TYPE;
(*
```

5.5.1 Editorial Comments

1. This has been made into a SUPERTYPE and renamed LEP PASSAGE to avoid conflict with the Feature Model.
2. The IDEF entity PASSAGE DEPOSITION SEQUENCE/P9 does not appear in this model. The information is conveyed via the attribute IS SURFACED BY.
3. The IDEF entity ELECTRICAL PLATED PASSAGE/P11 does not appear. The SIGNAL has been made an optional attribute of PASSAGE.

5.6 DEPOSITION

The specification of a deposited MATERIAL.

```
*)
ENTITY DEPOSITION;
  SPECIFICATION            : UNDEFINED;
  MATERIAL_SPECIFICATION   : LEP_MATERIAL;
  THICKNESS                : REAL;
WHERE
END_ENTITY;
(*
```

5.7 LAYER PASSAGE

The occurrence of a PASSAGE through a LAYER.

```
*)
ENTITY LAYER_PASSAGE SUBTYPE OF (LEP_PASSAGE);
  BOUNDARY : LOOP;
WHERE
END_ENTITY;
(*
```

5.8 COMPONENT LEAD PASSAGE

The occurrence of a PASSAGE intended for a COMPONENT LEAD. The PASSAGE may be plated.

```
*)
ENTITY COMPONENT_LEAD_PASSAGE SUBTYPE OF (LEP_PASSAGE);
WHERE
END_ENTITY;
(*
```

5.9 VIA

An ELECTRICAL PLATED PASSAGE whose sole purpose is to electrically connect LAYER ELEMENTS on two or more LAYERS. It cannot be associated with a COMPONENT LEAD.

```
*)
ENTITY VIA SUBTYPE OF (LEP_PASSAGE);
  VIA_NUMBER     : NUMBER;
  VIA_TYPE       : UNDEFINED;
WHERE
  SIGNAL <> NULL;
  IS_SURFACED_BY <> NULL;
END_ENTITY;
(*
```

PROPOSITIONS:

1. SIGNAL must not be NULL
2. The PASSAGE must be plated

5.10 LAYER TO LAYER JOIN STRING

The occurrence of a PASSAGE as a means of achieving one or more LAYER TO LAYER JOINS.

```
*)
ENTITY LAYER_TO_LAYER_JOIN_STRING SUBTYPE OF (LEP_PASSAGE);
  JOIN_PATH                    : PATH;
WHERE
END_ENTITY;
(*
```

5.11 LAYER DEPOSITION

A LAYER whose LAYER ELEMENTS are implemented as an additive process.

```
*)
ENTITY LAYER_DEPOSITION;
  IS_BASED_ON                 : LAYER;
  DEPOSITION_SPECIFICATION : DEPOSITION;
WHERE
END_ENTITY;
(*
```

6 RELATIONSHIP WITH OTHER EXPRESS MODELS

This Section provides some new EXPRESS models that are required by the LEP model. These are not fully defined but appear as dotted boxes in the LEP IDEF model.

6.1 LEP MATERIAL

The specification of a material as used in the LEP model.

MATERIAL is the tangible substance out of which a product is made. A material is made of exactly one constituted (mixture of) chemical compound possessing consistent characteristics and properties.

```
*)
ENTITY LEP_MATERIAL;
  MATERIAL_NAME             : STRING;
  MATERIAL_SPECIFICATION   : UNDEFINED;
  ELECTRICAL_CONDUCTIVITY : REAL;
  THERMAL_CONDUCTIVITY     : REAL;
END_ENTITY;
(*
```

6.1.1 Editorial Comments

1. This has been renamed LEP MATERIAL to distinguish from other MATERIAL definitions.

6.2 DEF ELEC LOG LINK

This defines a LEP electrical signal.

```
*)
ENTITY DEF_ELEC_LOG_LINK;
  SIGNAL_ATTRIBUTES : UNDEFINED;
END_ENTITY;
(*
```

6.3 LEP COMPONENT

A COMPONENT is an individual part or combination of parts that performs a specific design function(s).

A LEP COMPONENT is a type of product which is identified as being part of the next assembly to the subject of the LEP model. As such, it has two or more COMPONENT LEADs. It can either be mounted on a LAYERED ELECTRICAL PRODUCT in a subsequent assembly process or be implemented as a LAYER ELEMENT of the product.

```
*)
ENTITY LEP_COMPONENT;
  COMPONENT_DESCRIPTION : STRING;
  SPECIFICATION         : UNDEFINED;
  HEIGHT                : REAL;
  WIDTH                 : REAL;
  LENGTH                : REAL;
  LEADS                 : SET [2 : #] OF COMPONENT_LEAD;
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

COMPONENT_DESCRIPTION The nomenclature commonly recognized as an identifier of a component. Examples are capacitor, DIP etc.

SPECIFICATION The specification of the component. For example a reference to a catalogue number or part number.

HEIGHT The vertical displacement of the top of the unit from an installation surface.

WIDTH The installed horizontal width of the component body.

LENGTH The installed horizontal length of the component body.

LEADS The set of connections on the component.

6.4 COMPONENT LEAD

A connection "point" on a component. That is, an element of a LEP COMPONENT that serves as a mechanical and/or electrical connector in an assembly.

```
*)
ENTITY COMPONENT_LEAD;
  LEAD_ID         : INTEGER;
  LEAD_MAX_WIDTH  : REAL;
  SIGNAL          : OPTIONAL DEF_ELEC_LOG_LINK;
  ABSTRACTED_AS   : OPTIONAL VERTEX;
END_ENTITY;
(*
```

ATTRIBUTE DEFINITIONS:

LEAD_ID The "pin number" of the lead.

LEAD_MAX_WIDTH The diameter of a round lead or the diagonal of a rectangular lead.

SIGNAL The electrical signal carried by the lead

ABSTRACTED_AS An abstraction of the location of the lead.

6.5 IDEF and EXPRESS Model Relationships

The following entities are defined in the IDEF model which already exist, or have equivalents, in the EXPRESS IPIM. These entity are not included in the above EXPRESS model.

IDEF Entity	IPIM Entity
VERTEX/SS11	VERTEX
EDGE/SS9	EDGE
POINT/SS17	CARTESIAN_TWO_COORDINATE
FACE/SS2	FACE
LOOP/SS7	LOOP
PATH/SS29	PATH
EDGE PATH STRUCTURE/SS30	EDGE LOGICAL STRUCTURE
EDGE LOOP STRUCTURE/SS10	EDGE LOGICAL STRUCTURE
SURFACE/SS1	SURFACE
TRIMMED SURFACE/SS5	FACE
LOOP STRUCTURE	LOOP LOGICAL STRUCTURE
DIRECTION/SS33	THREE SPACE DIRECTION
CURVE/SS13	CURVE
CIRCLE/SS16	CIRCLE
LINE/SS15	LINE
CURVE SEGMENT STRUCTURE	EDGE

6.6 Other EXPRESS Entities

The following existing entities in the EXPRESS IPIM have been used in this model. Their definitions have not been repeated here.

NAME	DESCRIPTION
SIZE	A toleranced REAL value
AXIS PLACEMENT	A local coordinate system

7 Classification Scheme

The following indented list provides the classification scheme for the Layered Electrical Product Model.

- COMPONENT LEAD
- DEF ELEC LOG LINK
- DEPOSITION
- ICON
- ICON PLACEMENT
- LAYER
- LAYER DEPOSITION
- LAYER ELEMENT
- LAYER ELEMENT SUBREGION
  - ELEMENT JOIN SUBREGION
- LAYERED ELECTRICAL PRODUCT
- LEP COMPONENT
- LEP JOIN
  - ASSEMBLY JOIN
    - COMPONENT LEAD JOIN
    - LAYER TO LAYER JOIN
  - INTRALAYER JOIN
- LEP MATERIAL
- LEP PASSAGE
  - COMPONENT LEAD PASSAGE
  - LAYER PASSAGE
  - LAYER TO LAYER JOIN STRING
  - VIA
- LEP SHAPE
  - AREA SHAPE
  - GRAPH SHAPE
    - CLOSED GRAPH SHAPE
    - OPEN GRAPH SHAPE
  - POINT SHAPE
    - EXPLICIT POINT SHAPE
    - IMPLICIT POINT SHAPE
- LOGICAL SUBREGION
  - INFORMATION SUBREGION
  - RESTRICTED SUBREGION
- PANEL
- PANEL DETAIL
- PANEL LAYERED ELECTRICAL PRODUCT
- TEST COUPON
- TEXT
- TEXT PLACEMENT

8 Undefined Entities

An EXPRESS model must be *fully refined.* That means that there must be a trace from every definition to one of the EXPRESS "primitives" — NUMBER (Real or Integer), LOGICAL, STRING, ENUMERATION.

From a refining viewpoint there are three classes of entities in an EXPRESS model:

INCOMPLETE Those entities which have one or more attributes that are UNDEFINED.

PARTIALLY REFINED Those entities which have no UNDEFINED attributes but where one or more of the attributes refer, either directly or via a chain of entities, to an INCOMPLETE entity.

REFINED Those entities whose every attribute can be traced to an EXPRESS primitive.

Table 1 indicates which entities in the model are REFINED, PARTIALLY REFINED or INCOMPLETE. This classification is independent of whether the entity actually represents what it is intended to.

If the incomplete entities in the model were fully refined, then all the entities would automatically become fully refined. Note that the model is, in some sense, complete in EXPRESS terms because of the use of the UNDEFINED entity type for those attributes that have not yet been specified. In most cases the incomplete entities (or attributes of these) are integration points with other models yet to be specified.

ENTITY	REFINED	PARTIALLY REFINED	INCOMPLETE
COMPONENT LEAD		X	
DEPOSITION			X
DEF ELEC LOG LINK			X
LEP JOIN	X		
ASSEMBLY JOIN	X		
COMPONENT LEAD JOIN	X		
LAYER TO LAYER JOIN	X		
INTRALAYER JOIN	X		
LEP SHAPE	X		
AREA SHAPE	X		
GRAPH SHAPE	X		
CLOSED GRAPH SHAPE	X		
OPEN GRAPH SHAPE	X		
POINT SHAPE	X		
EXPLICIT POINT SHAPE	X		
IMPLICIT POINT SHAPE			X
ICON			X
ICON PLACEMENT		X	
LAYER			X
LAYER DEPOSITION		X	
LAYER ELEMENT		X	
LAYER ELEMENT SUBREGION	X		
ELEMENT JOIN SUBREGION	X		
LAYERED ELECTRICAL PRODUCT			X
LEP COMPONENT			X
LEP MATERIAL			X
LOGICAL SUBREGION	X		
INFORMATION SUBREGION		X	
RESTRICTED SUBREGION	X		
PANEL		X	
PANEL DETAIL		X	
PANEL LAYERED ELECTRICAL PRODUCT		X	
LEP PASSAGE		X	
COMPONENT LEAD PASSAGE		X	
LAYER PASSAGE		X	
LAYER TO LAYER JOIN STRING		X	
VIA			
TEST COUPON			X
TEXT			X
TEXT PLACEMENT		X	

Table 1: Entity refinement status

NOTE: The following Resource Entities are taken from the Wilson-Kennicott model. They have been utilized by the Cal Poly Task Team Extension to construct the Shape/Size section of the Layered Electrical Product Model. Note that the *Shell* and *Region* entities were not referenced by the Team because the Team was concerned with the properties of *layers* which could be adequately described using the Vertex, Edge, Path, Loop and Face entities.

APPENDIX A8

SAMPLE DATABASE TABLES

March 18, 1988

A.8 SAMPLE DATABASE

The CPTTX Demo Program is a simple database implementation of the data model. A copy of the current version of this demonstration database can be downloaded from the National Bureau of Standards Bulletin Board at (301)-963-6234. Written in dBASE III Plus, the demo program runs on an IBM PC/XT/AT and PS2 or compatible computer equiped with a graphics adapter (CGA/EGA/VGA).

A.8.1 IMPLEMENTATION

In the demo database, each entity type becomes a relation (dBASE table), and each instance of the entity becomes the rows (dBASE records), and the attributes of the entity become the columns (dBASE fields) of the relation. Therefore, the database remains as faithful as it can be to the data model.

The dBASE tables were populated with data from the USAF PDDI extension project. About 500K of data in Step file format were processed and imported directly into dBASE data files. Figure A.8.1-1 is a picture of the three-layer board used in the demo which shows all pads and traces. Figure A.8.1-2 displays all components on that board.

A.8.2 SAMPLE DATABASE STRUCTURES

The following dBASE structures describe each entity and its dBASE table filename, definition of each field, index filename, and key field(s). All tables are indexed by their key field(s).

DATABASE STRUCTURE

AREA SHAPE / C15

Structure for database : AREA_S.DBF

Field	Field name	Type	Width	Dec
1	S_ID	Numeric	5	
2	FACE_ID	Numeric	5	
** Total **			11	

AREA_S.NDX -- Indexed on: s_id

COMPONENT / S6

Structure for database : COMPO.DBF

Field	Field name	Type	Width	Dec
1	COMP_ID	Numeric	5	
2	REF_DES	Character	5	
3	DESC	Character	30	
** Total **			41	

COMPO.NDX -- Indexed on: comp_id

COMPONENT LEAD / S7

Structure for database : COMPO_L.DBF

Field	Field name	Type	Width	Dec
1	COMP_ID	Numeric	5	
2	PIN_ID	Numeric	5	
3	VERTEX_ID	Numeric	5	
** Total **			16	

COMPO_L.NDX -- Indexed on: comp_id

COMPONENT LEAD JOIN / C10

Structure for database : COMPO_S.DBF

Field	Field name	Type	Width	Dec
1	COMP_ID	Numeric	5	
2	S_ID	Numeric	5	
** Total **			11	

COMPO_S.NDX -- Indexed on: comp_id

CIRCLE / SS16

Structure for database : CIRCLE.DBF

Field	Field name	Type	Width	Dec
1	CURVE_ID	Numeric	5	
2	CENTER_P	Numeric	5	
3	DIR	Character	5	
4	RADIUS	Numeric	6	4
** Total **			22	

CIRCLE.NDX -- Indexed on: curve_id

CURVE / SS13

Structure for database : CURVE.DBF

Field	Field name	Type	Width	Dec
1	CURVE_ID	Numeric	5	
2	CURVE_TYPE	Numeric	1	
3	NOTE	Character	10	
** Total **			17	

CURVE.NDX -- Indexed on: curve_id

CURVE SEGMENT STRUCTURE / SS14

Structure for database : CURVE_S.DBF

Field	Field name	Type	Width	Dec
1	EDGE_ID	Numeric	5	
2	CURVE_ID	Numeric	5	
** Total **			11	

CURVE_S.NDX -- Indexed on: edge_id

DEFINED ELECTRICAL LOGICAL LINK / S5

Structure for database : DELL.DBF

Field	Field name	Type	Width	Dec
1	LINK_ID	Numeric	5	
2	SIGNAL_NAM	Character	10	
** Total **			16	

DELL.NDX -- Indexed on: link_id

EDGE 1/ SS9

Structure for database : EDGE.DBF

Field	Field name	Type	Width	Dec
1	EDGE_ID	Numeric	5	
2	VERTEX_ID1	Numeric	5	
3	VERTEX_ID2	Numeric	5	
** Total **			16	

EDGE.NDX -- Indexed on: edge_id

EXPLICIT ELEMENT POINT SHAPE / C31

Structure for database : EEPS.DBF

Field	Field name	Type	Width	Dec
1	S_ID	Numeric	5	
2	FACE_ID	Numeric	5	
** Total **			11	

EEPS.NDX -- Indexed on: s_id

ELECTRICAL LAYER ELEMENT / C7

Structure for database : ELE.DBF

Field	Field name	Type	Width	Dec
1	PDPI_ID	Numeric	5	
2	V_ID	Numeric	5	
3	L_ID	Numeric	5	
4	LE_ID	Numeric	5	
5	LINK_ID	Numeric	5	
** Total **			26	

ELE.NDX -- Indexed on: link_id

EDGE LOOP STRUCTURE / SS10

Structure for database : ELS.DBF

Field	Field name	Type	Width	Dec
1	LOOP_ID	Numeric	5	
2	EDGE_ID	Numeric	5	
3	EL_ORIENT	Character	30	
** Total **			41	

ELS.NDX -- Indexed on: loop_id

EDGE PATH STRUCTURE / SS30

Structure for database : EPS.DBF

Field	Field name	Type	Width	Dec
1	PATH_ID	Numeric	5	
2	EDGE_ID	Numeric	5	
** Total **			11	

EPS.NDX -- Indexed on: path_id+edge_id

FACE / SS2

Structure for database : FACE.DBF

Field	Field name	Type	Width	Dec
1	FACE_ID	Numeric	5	
2	DESC	Character	50	
** Total **			56	

FACE.NDX -- Indexed on: face_id

LAYERED ELECTRICAL PRODUCT / C1

Structure for database : LEPIV.DBF

Field	Field name	Type	Width	Dec
1	PDPI_ID	Numeric	5	
2	V_ID	Numeric	5	
3	FACE_ID	Numeric	5	
4	LEP_TYPE	Character	5	
5	COMMENT	Character	40	
** Total **			61	

LEPIV.NDX -- Indexed on: pdpi_id+v_id

LAYERED ELEMENT SHAPE / C28

Structure for database : LES.DBF

Field	Field name	Type	Width	Dec
1	S_ID	Numeric	5	
2	PDPI_ID	Numeric	5	
3	V_ID	Numeric	5	
4	L_ID	Numeric	5	
5	LE_ID	Numeric	5	
** Total **			26	

LES.NDX -- Indexed on: pdpi_id+v_id+l_id+le_id

```
---------------------------------------------------------
GRAPH SHAPE / C16
Structure for database : LINE_S.DBF
Field          Field name       Type          Width     Dec

1              S_ID             Numeric       5
2              LS_TYPE          Numeric       5
3              WIDTH            Numeric       19        15
** Total **               30

LINE_S.NDX -- Indexed on: s_id
```

```
---------------------------------------------------------
LOOP / SS7
Structure for database : LOOP.DBF
Field          Field name       Type          Width     Dec

1              LOOP_ID          Numeric       5
2              DESC             Character     50
**  Total  **                                 56

LOOP.NDX -- Indexed on: loop_id
```

```
---------------------------------------------------------
LOOP STRUCTURE / SS8
Structure for database : LOOP_S.DBF
Field          Field name       Type          Width     Dec

1              FACE_ID          Numeric       5
2              LOOP_ID          Numeric       5
3              LS_ORIENT        Character     30
**  Total  **                                 41

LOOP_S.NDX – Indexed on: face_id
```

```
---------------------------------------------------------
OPEN GRAPH SHAPE / C35
Structure for database : OLINE_S.DBF
Field          Field name       Type          Width     Dec

1              S_ID             Numeric       5
2              PATH_ID          Numeric       5
**  Total  **                                 11

OLINE_S.NDX -- Indexed on: s_id
```

PATH / SS29

Structure for database : PATH.DBF

Field	Field name	Type	Width	Dec
1	PATH_ID	Numeric	5	
** Total **			6	

PATH.NDX -- Indexed on: path_id

POINT / SS17

Structure for database : POINT.DBF

Field	Field name	Type	Width	Dec
1	POINT_ID	Numeric	5	
2	PDDI_P_ID	Numeric	5	
** Total **			11	

POINT.NDX -- Indexed on: point_id

POINT SHAPE / C17

Structure for database : POINT_S.DBF

Field	Field name	Type	Width	Dec
1	S_ID	Numeric	5	
2	ASPECT	Character	10	
3	VERTEX_ID	Numeric	5	
** Total **			21	

POINT_S.NDX -- Indexed on: s_id

PDDI POINT -- this is the XYZ coordinate mapping to PDDI project data

Structure for database : POINT1.DBF

Field	Field name	Type	Width	Dec
1	POINT_ID	Numeric	5	
2	X	Numeric	19	15
3	Y	Numeric	19	15
4	Z	Numeric	19	15
** Total **			63	

POINT1.NDX -- Indexed on: point_id

```
---------------------------------------------------------------
SHAPE  /  C32
Structure for database : S.DBF
Field        Field name        Type        Width        Dec

1            S_ID              Numeric     5
2            S_TYPE            Numeric     5
** Total **                                1 1

S.NDX -- Indexed on: s_id
```

A.8.3 DATA DICTIONARY

The following table gives descriptions and definitions for all fields (attributes), key as well as non-key, in the demo database.

FIELD NAME	TYPE	LEN	DEC	DATABASE	DEFINITION
ASPECT	C	10	0	POINT_S.DBF	Aspect Ratio
CENTER_P	N	5	0	CIRCLE.DBF	Center point coord.
COMMENT	C	40	0	LEPIV.DBF	Comment
COMP_ID	N	5	0	COMPO_L.DBF COMPO.DBF COMPO_S.DBF	Component ID
CURVE_ID	N	5	0	CURVE_S.DBF CURVE.DBF CIRCLE.DBF	Curve ID
CURVE_TYPE	N	1	0	CURVE.DBF	Curve Type
DESC	C	50	0	FACE.DBF LOOP.DBF COMPO.DBF	Description
DIR	C	5	0	CIRCLE.DBF	Direction
EDGE_ID	N	5	0	ELS.DBF EDGE.DBF CURVE_S.DBF EPS.DBF	Edge ID
EL_ORIENT	C	30	0	ELS.DBF	Element Orientation
FACE_ID	N	5	0	AREA_S.DBF LOOP_S.DBF LEPIV.DBF FACE.DBF EEPS.DBF	Face ID
LEP_TYPE	C	5	0	LEPIV.DBF	Layered Elec. Prod. Type
LE_ID	N	5	0	ELE.DBF LES.DBF	Layer Element ID
LINK_ID	N	5	0	DELL.DBF ELE.DBF	Link ID
LOOP_ID	N	5	0	LOOP_S.DBF ELS.DBF LOOP.DBF	Loop ID
LS_ORIENT	C	30	0	LOOP_S.DBF	Loop Struc. Orientation
LS_TYPE	N	5	0	LINE_S.DBF	Loop Structure Type
L_ID	N	5	0	ELE.DBF LES.DBF	Layer ID
NOTE	C	10	0	CURVE.DBF	Curve Note
PATH_ID	N	5	0	OLINE_S.DBF PATH.DBF EPS.DBF	Path ID
PDDI_P_ID	N	5	0	POINT.DBF	PDDI Point ID

FIELD NAME	TYPE	LEN	DEC	DATABASE	DEFINITION
PDPI_ID	N	5	0	LEPIV.DBF ELE.DBF LES.DBF	Physically Defined Product Item ID
PIN_ID	N	5	0	COMPO_L.DBF	Component Pin ID
POINT_ID	N	5	0	POINT1.DBF POINT.DBF	Point ID
RADIUS	N	6	4	CIRCLE.DBF	Circle Radius
REF_DES	C	5	0	COMPO.DBF	Component Reference Designator
SIGNAL_NAM	C	0	0	DELL.DBF	Signal Name
S_ID	N	5	0	AREA_S.DBF LES.DBF S.DBF POINT_S.DBF EEPS.DBF COMPO_S.DBF LINE_S.DBF OLINE_S.DBF	Shape ID
S_TYPE	N	5	0	S.DBF	Shape Type
VERTEX_ID	N	5	0	POINT_S.DBF COMPO_L.DBF	Vertex ID
VERTEX_ID1	N	5	0	EDGE.DBF	Vertex ID 1
VERTEX_ID2	N	5	0	EDGE.DBF	Vertex ID 2
V_ID	N	5	0	LEPIV.DBF ELE.DBF LES.DBF	Version ID
WIDTH	N	19	15	LINE_S.DBF	Line Width
X	N	19	15	POINT1.DBF	X Coordinate
Y	N	19	15	POINT1.DBF	Y Coordinate
Z	N	19	15	POINT1.DBF	Z Coordinate

ALL LAYERS

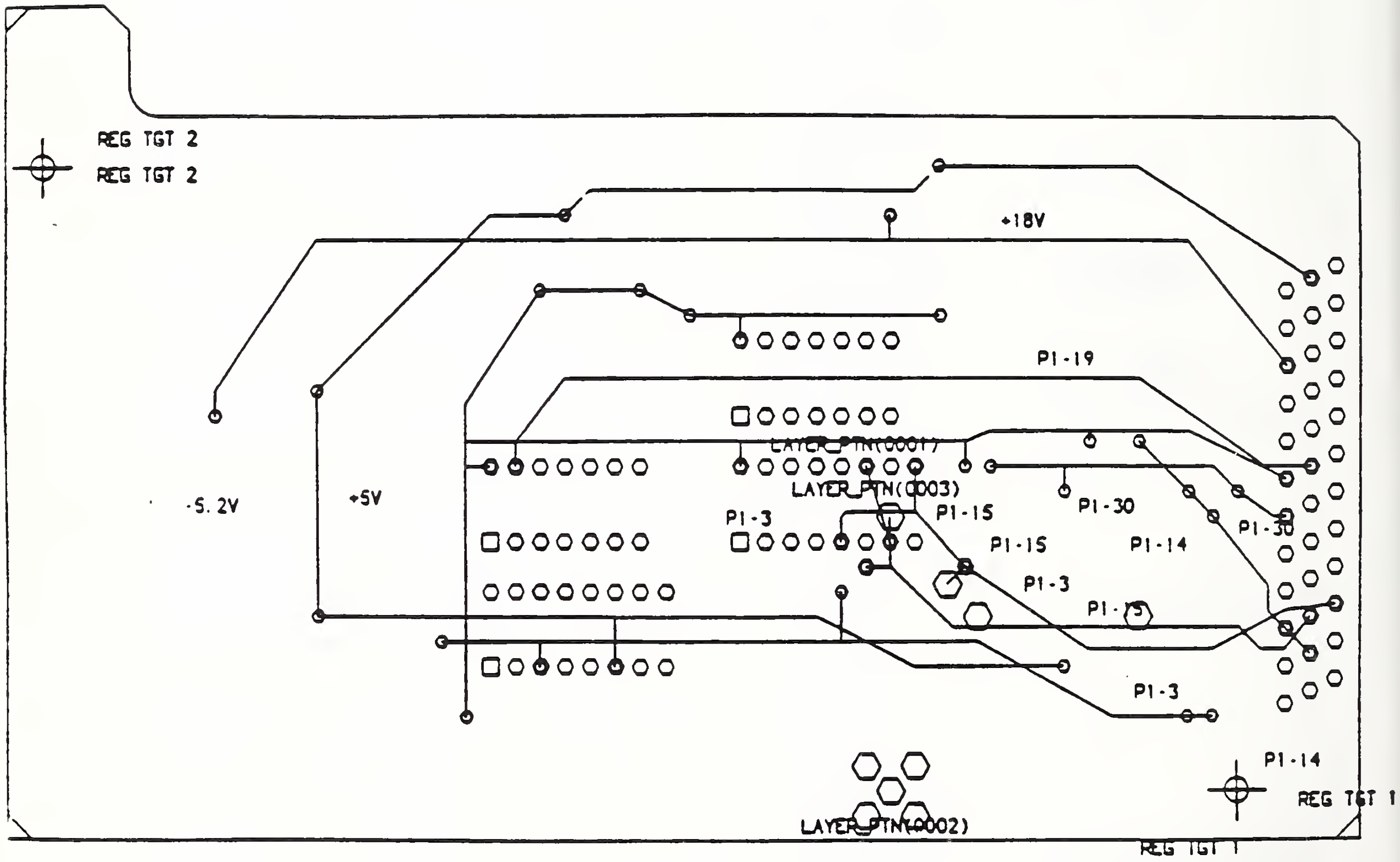


Figure A.8.1-1

COMPONENTS

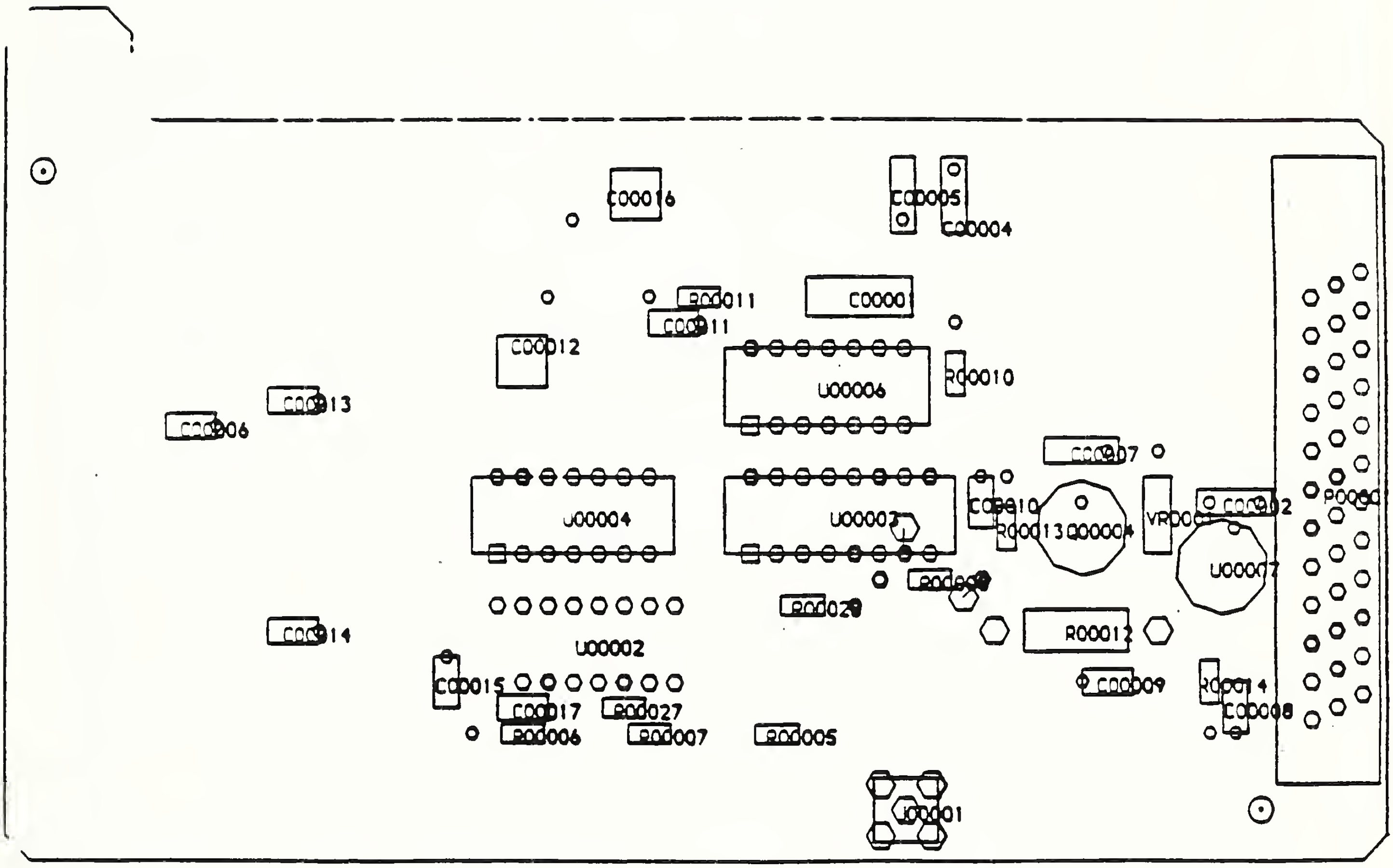


Figure A.8.1-2

APPENDIX A9

QUERY EXAMPLES

March 18, 1988

A.9 SAMPLE QUERY

A.9.1 QUERY TO DISPLAY PWB OUTLINE

The reference path analysis for displaying the board outline is as follows:

```
lep    face        loop         edge   vertex      point
C1  -  SS2  -  SS8  -  SS7  -  SS10  -  SS9  -  SS11  -   SS17        STOP POINT
                                 |
                                 |     curve    line
                               SS14 - SS13 -.  SS15                  LINE
                                                 |
                                                 - -  SS17           START POINT
                                                 |
                                               SS33                  DIRECTION
```

The following is a commented listing of an actual dBASE session to display a board outline.

<u>NOTATION CONVENTION:</u>

C>	is the DOS prompt.
dBASE>	is the dBASE prompt.
. . .	vertical ellipsis indicates that a portion of the data listing is omitted.
*	comments.

```
* Start dBASE
C>dbase

* Use the table (entity) Layered Electronic Product. We first have to find out
* what version of what product for which we want to display the outline.
dBASE> use lepiv index lepiv

* List all records (instances) of this entity
dBASE> list all
Record#     PDPI_ID     V_ID       FACE_ID     LEP_TYPE    COMMENT
      2     0           1          0           PWB
      1     1           1          1           PWB         Version 1 - CURRENT
      3     2           5          2           PWB
      4     3           7          3           PWB
```

```
* Suppose we want version 1 of product 1. Let's find that record.
dBASE> locate for PDPI_ID = 1 .and. V_ID = 1
Record = 1

* dBASE tells us that record number 1 satisfied our condition of PDPI_ID = 1
* and V_ID = 1 and that the face id of this record is 1. Let's save this
* non-key attribute FACE_ID to a variable called our_key.
dBASE> our_key = FACE_ID
   1

* Find out all we can about this face.
dBASE> use face index face
dBASE> list all
Record#        FACE_ID              DESC
    1          1                    LEP Material Bound
    2          2                    L1 Material Bound
    3          3                    L2 Material Bound
    4          4                    L3 Material Bound
    5          5                    Component Outline
    .          .                    .
    .          .                    .
    .          .                    .
   38          38                   Component Outline
   39          39                   Component Outline
   40          40                   Component Outline
   41          41                   Pad Outline
   42          42                   Pad Outline
   43          43                   Pad Outline
   44          44                   Pad Outline
   45          45                   Pad Outline
   46          46                   Pad Outline
   47          47                   Pad Outline
   48          48                   Pad Outline

* Locate the face record that matches our saved key.
dBASE> locate for FACE_ID = our_key
Record =      1
dBASE> display
Record#        FACE_ID              DESC
    1          1                    LEP Material Bound

* A face is bounded by a loop. Find that loop.
dBASE> use loop_s index loop_s
dBASE> list all
Record# FACE_ID   LOOP_ID       LS_ORIENT
    1          1   1            Outer
   2            2       2               Outer
   3            3       3               Outer
   4            4       4               Outer
   5            5       5               Outer
   .            .       .               .
   .            .       .               .
   .            .       .               .
   52           53      45              Inner
   53           54      46              Inner
   54           55      47              Inner
   55           56      48              Inner

dBASE> locate for FACE_ID = our_key
Record =      1
dBASE> display
Record#      FACE_ID      LOOP_ID      LS_ORIENT
   1         1            1            Outer

* The loop that defined our face has a loop_id of 1. Save this id.
dBASE> our_key = LOOP_ID
  1

* But since face has one or many relationship (P) with loop
*. structure, this face id may be bounded by more than 1 loop id so
* continue to locate further records that would have FACE_ID = our_key
dBASE> continue
End of LOCATE scope

* Apparently, this face is bounded by exactly one loop
dBASE> use loop index loop
dBASE> list all
Record#      LOOP_ID      DESC
   1         1            LEP Material Bound
   2         2            L1 Material Bound
   3         3            L2 Material Bound
   4         4            L3 Material Bound
   5         5
   .        -.            .
   .         .            .
   .         .            .
   53        54
   54        55
   55        56

* Bounded by what loop ?
dBASE> locate for LOOP_ID = our_key
Record =      1
dBASE> display
```

```
Record#     LOOP_ID     DESC
    1       1           LEP Material Bound

* Now that we know what loop it is, the next question is what edges
* participate in this loop
dBASE>use els index els
dBASE>list all
Record#     LOOP_ID     EDGE_ID     EL_ORIENT
   181      0           357
   182      0           358
    1       1           1
    2       1           2
    3       1           3
    4       1           4
    5       1           5
    6       1           6
    7       1           7
    8       1           8
    9       1           9
   10       1           10
   11       1           11
   12       1           12
   13       1           13
   14       1           14
   15       1           15
   16       1           16
   17       1           17
   18       1           18
   19       2           19
   20       2           20
   21       2           21
   22       2           22
   23       3           23
   24       3           24
   25       3           25
   26       3           26
   27       4           27
   28       4           28
    .       .           .
    .       .           .
    .       .           .
   199      53          541
   200      54          542
   201      55          543
   202      56          544
```

```
dBASE>list all for LOOP_ID = our_key
Record#        LOOP_ID        EDGE_ID        EL_ORIENT
     1          1              1
     2          1              2
     3          1              3
     4          1              4
     5          1              5
     6          1              6
     7          1              7
     8          1              8
     9          1              9
    10          1             10
    11          1             11
    12          1             12
    13          1             13
    14          1             14
    15          1             15
    16          1             16
    17          1             17
    18          1             18

* Edges 1 to 18 participate in defining our loop. Each edge in turn
* would have associated start and stop vertices.
dBASE>use edge index edge
dBASE>list all
Record#        EDGE_ID        VERTEX_ID1     VERTEX_ID2
     1          1              1              2
     2          2              2              3
     3          3              3              4
     4          4              4              5
     5          5              5              6
     6          6              6              7
     7          7              7              8
     8          8              8              9
     9          9              9             10
    10         10             10             11
    11         11             11             12
    12         12             12             13
    13         13             13             14
    14         14             14             15
    15         15             15             16
    16         16             16             17
    17         17             17             18
    18         18             18              1
    19         19             19             20
    20         20             20             21
    21         21             21             22
    22         22             22             19
```

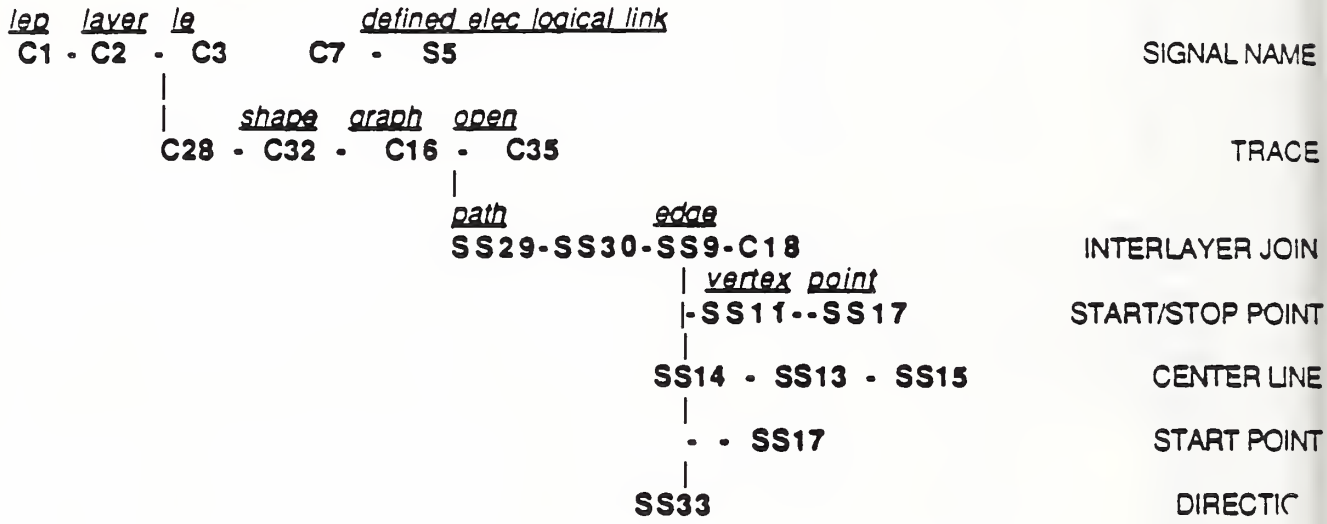
```
23        23        23        24
24        24        24        25
25        25        25        26
26        26        26        27
27        27        27        28
28        28        28        29
29        29        29        30
30        30        30        31
31        31        31        32
32        32        32        33
33        33        33        34
34        34        34        35
35        35        35        36
36        36        36        37
37        37        37        38
38        38        38        39
39        39        39        40
40        40        40        41
 .        .         .         .
 .        .         .         .
 .        .         .         .
538       539       425       425
539       540       426       426
540       541       427       427
541       542       428       428
542       543       429       429
543       544       430       430

dBASE>go top

* List all of our edges (edge id from 1 to 18)
dBASE>list all for EDGE_ID >= 1 .and. EDGE_ID <= 18
Record#   EDGE_ID   VERTEX_ID1  VERTEX_ID2
    1     1         1                2
    2     2         2                3
    3     3         3                4
    4     4         4                5
    5     5         5                6
    6     6         6                7
    7     7         7                8
    8     8         8                9
    9     9         9                10
   10     10        10               11
   11     11        11               12
   12     12        12               13
   13     13        13               14
   14     14        14               15
   15     15        15               16
```

```
    16          16          16               17
    17          17          17               18
    18          18          18               1

* These 18 edges bound the board outline. Each edge defines a start vertex
* and a stop vertex and can be a Circle (same start and stop vertices) or a
* Line (different start and stop vertices). We must next find out what it is
* through Curve Segment Structure / SS14, Curve / SS13, etc. However, we can
* take a short cut here by noting that all of our edges are of type Line
* (see listing above).
* The XYZ coordinate for a particular vertex id can be found by inspecting
* table POINT, field POINT_ID which is the same as VERTEX_ID.
dBASE>use vertex index vertex
dBASE>list all for VERTEX_ID >= 1 .and. VERTEX_ID <= 18
Record#     VERTEX_ID
   1         1
   2         2
   3         3
   4         4
   5         5
   6         6
   7         7
   8         8
   9         9
   10        10
   11        11
   12        12
   13        13
   14        14
   15        15
   16        16
   17        17
   18        18

* In the demo database implementation, the XYZ coordinate fields was replaced
* by a field called PDDI_P_ID which is the actual PDDI project point id to
* provide a mapping between our coordinate system and that of PDDI.
dBASE>use point index point
dBASE>list all for POINT_ID >= 1 .and. POINT_ID <= 18
Record#     POINT_ID    PDDI_P_ID
   11        1           331
   3         2           262
   12        3           333
   13        4           334
   14        5           335
   15        6           336
   16        7           337
   17        8           338
   18         9            339
   19         10           340
   20         11           341
   21         12           342
   22         13           343
   23         14           344
   4          15           263
   24         16           346
   6          17           265
   7          18           266

* At last, we can obtain XYZ coordinate information from the POINT1 table
* which is the PDDI point entity table. Let's get coordinates for point id
* from 333 to 340 since they have sequential id numbers.
dBASE>use point1 index point1
dBASE>list all for POINT_ID >= 333 .and. POINT_ID <= 340
Record#  POINT_ID                   X                       Y                        Z
248           333   -4.550000190734900   3.150000095367400    0.000000000000000
249           334   -4.449999809265100   3.049999952316300    0.000000000000000
250           335   -4.449999809265100   2.819999933242800    0.000000000000000
251           336   -4.447000026702900   2.792999982833900    0.000000000000000
252           337   -4.438000202179000   2.767999887466400    0.000000000000000
25            338   -4.423999786377000   2.744999885559100    0.000000000000000
254           339   -4.405000209808300   2.726000070571900    0.000000000000000
255           340   -4.381999969482400   2.711999893188500    0.000000000000000

* These XYZ coordinates then can be plotted to the display screen or on paper
*  for viewing.
```

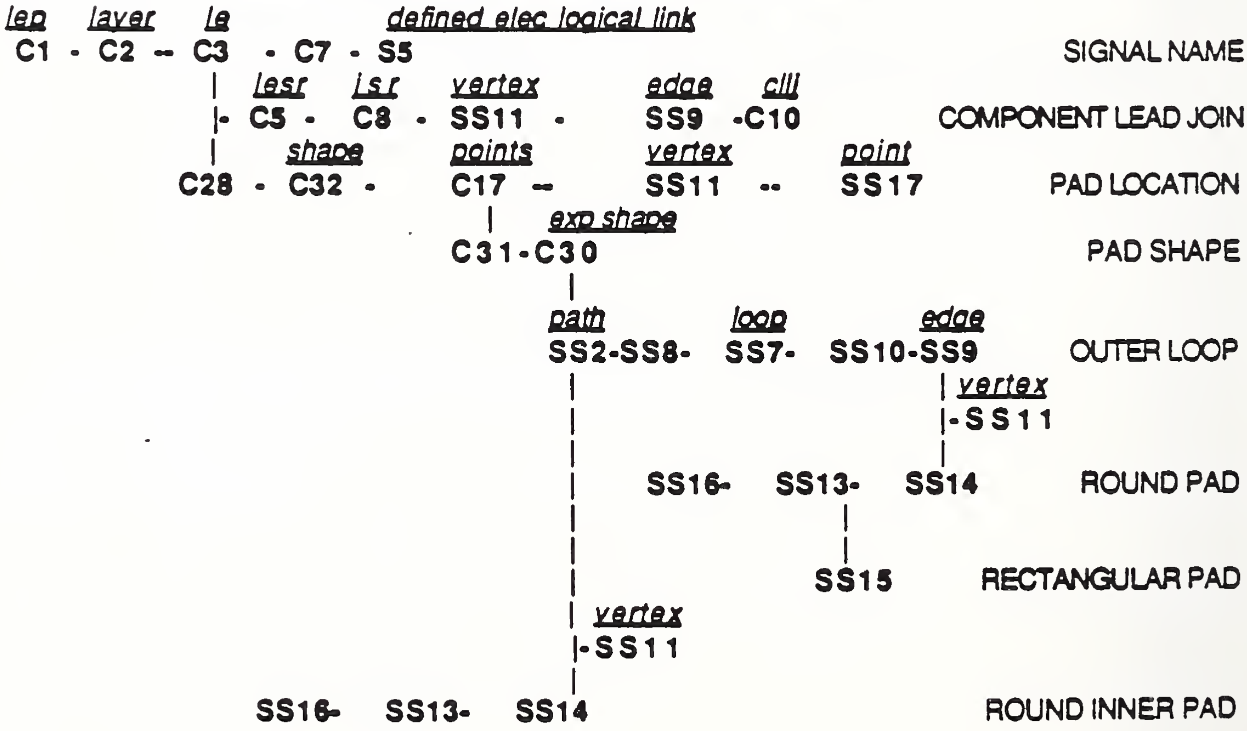
A.9.2 REFERENCE PATH ANALYSIS FOR ADDITIONAL QUERIES

Additional queries can be developed by using the reference path analysis as followed:

A.9.2.1 Display all traces for signal +5V on circuit layer 1

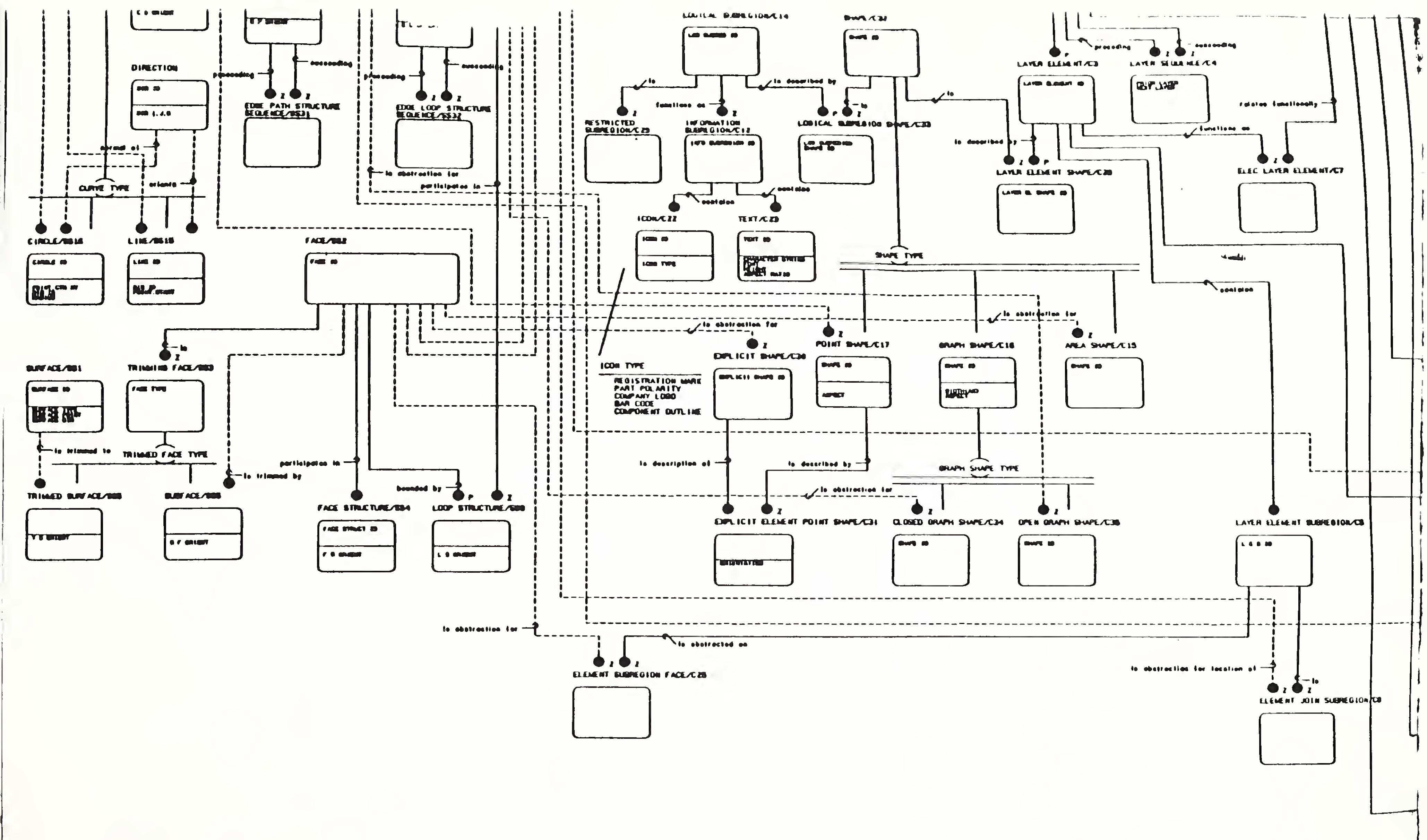


A.9.2.2 Display all component lead pads for signal +5V on circuit layer 1



A.9.2.2 Display all +5V signal vias including via hole

```
lep   layer   le            defined elec logical link
C1 - C2 -- C3   - C7 - S5                                      SIGNAL NAME
            |  lesr   eisr   vertex       edge   clli
            |- C5 -   C8  -  SS11  -      SS9  -C10            COMPONENT LEAD JOIN
            |                              |
            |                             C9  lljoin           LAYER TO LAYER JOIN
            |
            |    shape       points       vertex      point
          C28 - C32 -        C17  --      SS11   --   SS17     VIA PAD LOCATION
                              |   exp shape
                             C31-C30 -                         VIA PAD SHAPE
                                  |
                                 path       loop      edge
                                 SS2-SS8-   SS7-   SS10-SS9    OUTER LOOP
                                  |                    | vertex
                                  |                    |-SS11
                                  |                    |
                                  |    SS16-   SS13-   SS14    VIA PAD ROUND
                                  |             |
                                  |             |
                                  |            SS15            RECTANGULAR VIA
                                  | vertex
                                  |-SS11
                                  |
              SS16-   SS13-   SS14                             ROUND PLATED HOLE
```


DIRECTION
EDGE PATH STRUCTURE SEQUENCE/SS31
EDGE LOOP STRUCTURE SEQUENCE/SS32
preceding
succeeding
is abstraction for
participates in
CURVE TYPE
orients
CIRCLE/SS16
LINE/SS15
FACE/SS2
SURFACE/SS1
TRIMMING FACE/SS3
TRIMMED FACE TYPE
is trimmed to
is trimmed by
TRIMMED SURFACE/SS5
SURFACE/SS6
FACE STRUCTURE/SS4
LOOP STRUCTURE/SS9
bounded by
LOGICAL SUBREGION/C14
SHAPE/C32
RESTRICTED SUBREGION/C29
INFORMATION SUBREGION/C12
LOGICAL SUBREGION SHAPE/C30
functions as
is described by
contains
ICON/C22
TEXT/C23
ICON TYPE
REGISTRATION MARK
PART POLARITY
COMPANY LOGO
BAR CODE
COMPONENT OUTLINE
SHAPE TYPE
EXPLICIT SHAPE/C38
POINT SHAPE/C17
GRAPH SHAPE/C16
AREA SHAPE/C15
GRAPH SHAPE TYPE
is description of
EXPLICIT ELEMENT POINT SHAPE/C31
CLOSED GRAPH SHAPE/C34
OPEN GRAPH SHAPE/C35
LAYER ELEMENT/C3
LAYER SEQUENCE/C4
LAYER ELEMENT SHAPE/C20
ELEC LAYER ELEMENT/C7
relates functionally
LAYER ELEMENT SUBREGION/C5
ELEMENT SUBREGION FACE/C25
is abstracted as
is abstraction for location of
ELEMENT JOIN SUBREGION/C8

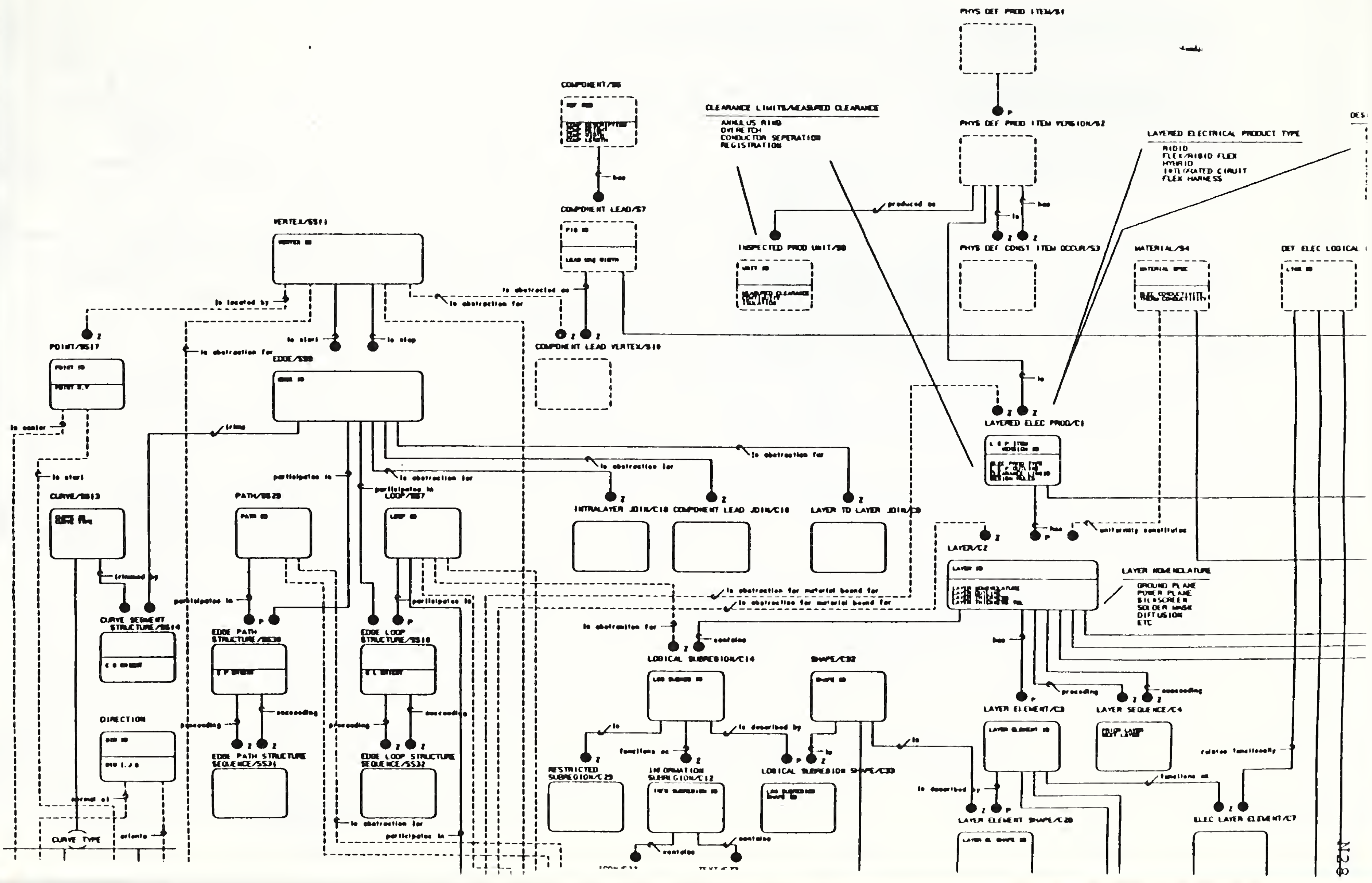
USED AT

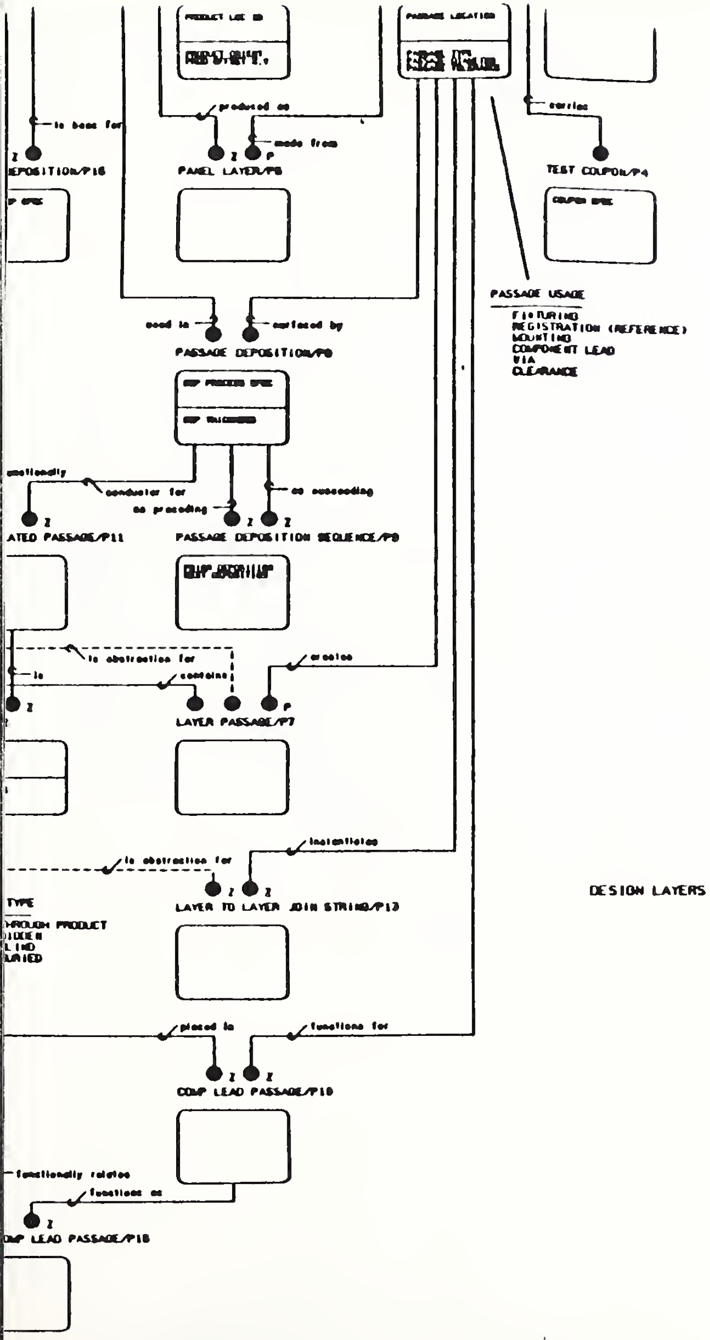
AUTHOR: SCOTT MADSEN

PROJECT: CAL POLY TASK TEAM EXTEND

EV:

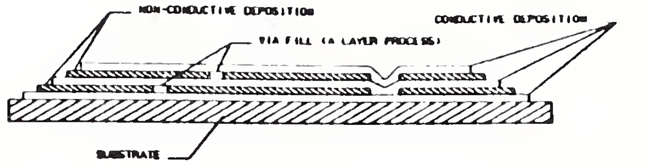
NOTES: 1 2 3 4 5 6 7 8 9 10



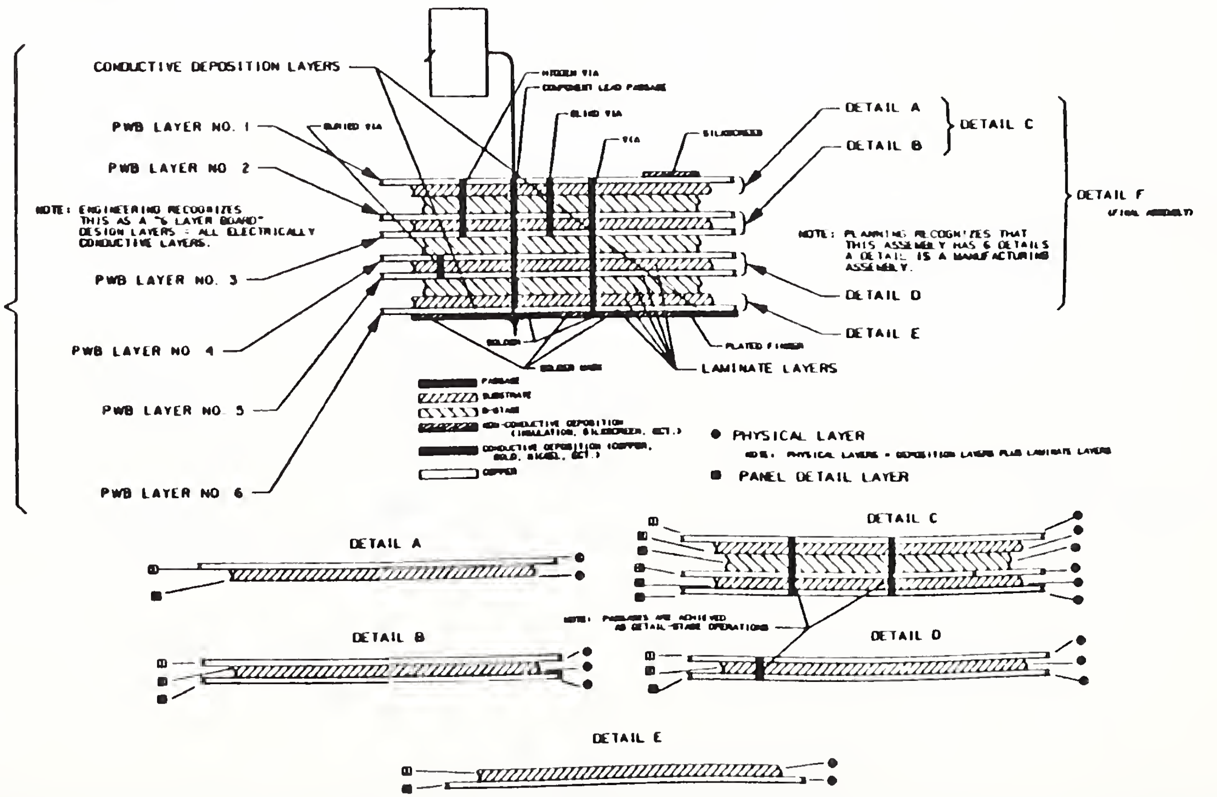


WAFER

HMA LAYERING TERMINOLOGY MAP



PWB LAYERING TERMINOLOGY MAP


CONDUCTIVE DEPOSITION LAYERS
PWB LAYER NO. 1
PWB LAYER NO 2
NOTE: ENGINEERING RECOGNIZES THIS AS A "6 LAYER BOARD" DESIGN LAYERS = ALL ELECTRICALLY CONDUCTIVE LAYERS.
PWB LAYER NO. 3
PWB LAYER NO 4
PWB LAYER NO. 5
PWB LAYER NO 6
DESIGN LAYERS
HIDDEN VIA
COMPONENT LEAD PASSAGE
BURIED VIA
BLIND VIA
VIA
SILKSCREEN
SOLDER
SOLDER MASK
PLATED FINGER
LAMINATE LAYERS
DETAIL A
DETAIL B
DETAIL C
DETAIL F
(FINAL ASSEMBLY)
NOTE: PLANNING RECOGNIZES THAT THIS ASSEMBLY HAS 6 DETAILS A DETAIL IS A MANUFACTURING ASSEMBLY.
DETAIL D
DETAIL E
PASSAGE
SUBSTRATE
B-STAGE
NON-CONDUCTIVE DEPOSITION (INSULATION, SILKSCREEN, ECT.)
CONDUCTIVE DEPOSITION (COPPER, GOLD, NICKEL, ECT.)
COPPER
● PHYSICAL LAYER
NOTE: PHYSICAL LAYERS = DEPOSITION LAYERS PLUS LAMINATE LAYERS
■ PANEL DETAIL LAYER
DETAIL A
DETAIL C
NOTE: PASSAGES ARE ACHIEVED AS DETAIL-STAGE OPERATIONS
DETAIL B
DETAIL D
DETAIL E

D ELECTRICAL PRODUCTS

NUMBER:

N2

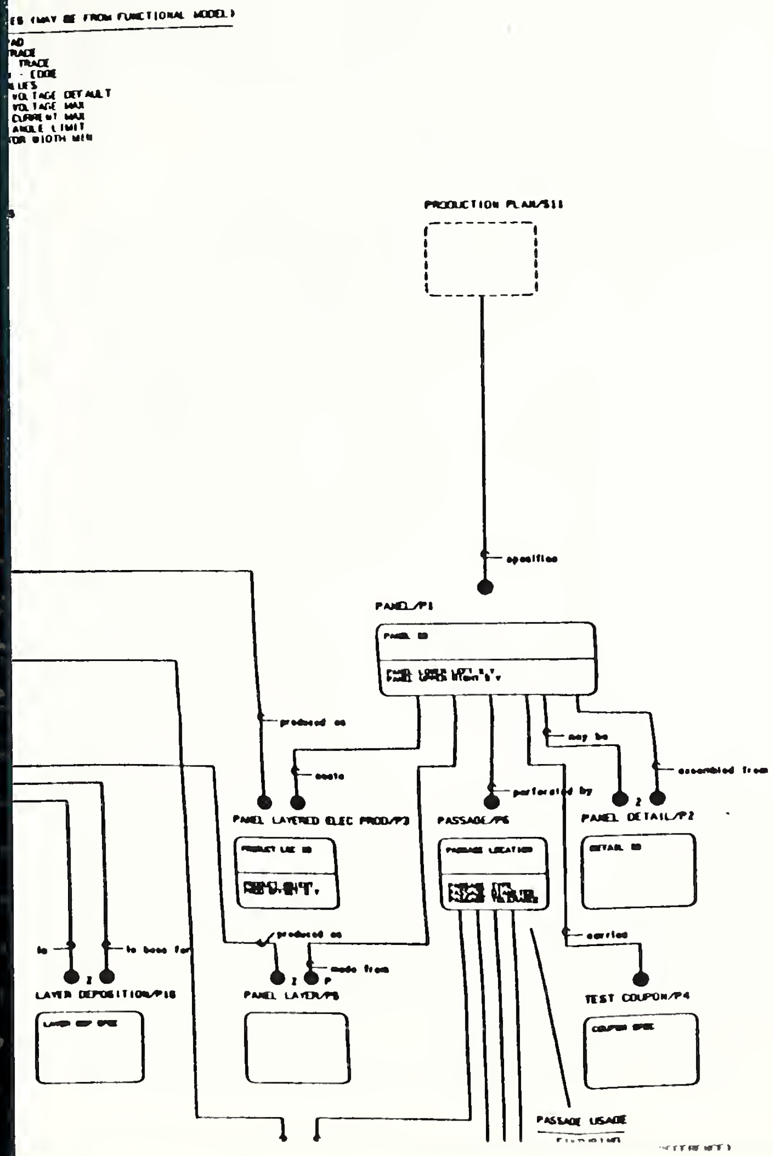
09/87

X

FT

OMMENDED

PUBLICATION



READERS:

This model version is the final draft of the Team following the meeting December 2 - 5 at NBS, Gaithersburg. There, final changes were made following a test population of a relational database constructed to this model. The keys as migrated by JANUS will not be included in this model overview but will be in the 19 view kit and final report to be presented in the January IGES/PDES meeting. After that meeting the chairman of the Electrical Applications Committee will coordinate further model activities, and reporting such work to the IEEE/DASS.

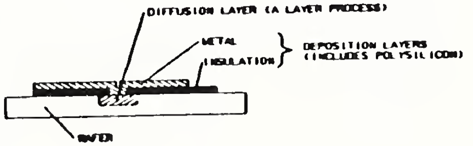
TEAM:

This version adds the Direction entity and the relation changes as agreed at Gaithersburg. Comments regarding this model may be telephoned to John Zimmerman, (816) 997-2932 prior to the IGES/PDES meeting. The stability achieved with the final version reflects the concern and dedication of the Team. Job well done!

CAL POLY TASK TEAM:

John Zimmerman	Allied Bendix	Team Modeler
Dzung Ha	McDonnell Douglas	Test DB Manager
George Goldsmith	McDonnell Douglas	Team Administrator
Mary Kennedy	McDonnell Douglas	Team Coordinator
Larry O'Connell	Sandia Labs	PDES Coordinator
Steven Platz	Unisys Corp	IEEE Coordinator
Roger Gale	DACOM Inc	IDEF Consultant
Noel Christensen	Allied Bendix	
Kazuo Hori	McDonnell Douglas	
William Loye	Loye Associates	
Scott Madsen	General Dynamics	
Robert Meagher	Eastman Kodak	
Paul Nelson	Hughes Corp	
Curtis Parks	General Dynamics	
Joan Tyler	NBS AMRF	

IC LAYERING TERMINOLOGY MAP


DIFFUSION LAYER (A LAYER PROCESS)
METAL
INSULATION
DEPOSITION LAYERS (INCLUDES POLYSILICON)
WAFER

HMA LAYERING TERMINOLOGY MAP

9 FEM Information Model

9.1 Purpose & Scope

This document defines and describes the IDEF1X reference model for engineering analysis using the finite element method. In addition to the IDEF1X data model there are two other matching models. One is in EXPRESS (section ?? for the complete EXPRESS listing) and the other is SML (section ?? for the complete SML listing).

9.1.1 Purpose

The purpose of this document is to propose an IDEF1X reference model along with the matching EXPRESS and SML models for the finite element method which is to be submitted to the ISO TC 184/SC4 and for inclusion in this standard.

9.1.2 Scope and Viewpoint

The scope of this document is limited to linearly elastic and thermal finite element models. The intent is to capture just the finite element model itself and none of the pre- and post processing information. Inclusion of the processing information, as well as non-linear material or geometric properties, is left to a future version of this specification.

The viewpoint of the committee creating this document is that of the finite element analyst with preference to structural mechanics.

9.1.3 FEM Model Fundamental Concepts and Assumptions

The FEM model fundamental concepts and assumptions are presented in this section in the form of rules. The following is a table of contents for these ideas.

SECTION 9: FEM INFORMATION MODEL

Concept	Rule Description
FEM/R-1	FEM Model and Id
FEM/R-2	FEM Element Characteristics
FEM/R-3	FEM Master Coordinate System
FEM/R-4	FEA
FEM/R-5	FEM Units
FEM/R-6	FEM Elements with Constant Geometric Properties
FEM/R-7	FEM Elements with Varying Geometric Properties
FEM/R-8	FEM Stress Recovery Coefficients
FEM/R-9	FEM Beam Shear Area
FEM/R-10	FEM Composite Laminate Z Direction
FEM/R-11	FEM Composite Laminate Z Offset
FEM/R-12	FEM Springs, Dampers, and Beams
FEM/R-13	FEM Anisotropic 2D and 3D Data (obsolete)
FEM/R-14	FEM Optional Shell Material Property Reference
FEM/R-15	Preventing Fatal Recursive Material ID Reference

FEM/R-1 FEM Model and ID

Any change to a FEM (e.g., changes to nodes, elements, groups, material properties, or geometric properties) causes a new model that must be issued a new FEM ID.

Date Created: 7th April 1985 by FEM Committee

FEM/R-2 FEM Element Characteristics

A finite element's characteristics are fully defined by three of the finite element's attributes. These attributes are FEM ELEMENT ORDER, FEM ELEMENT SHAPE, and FEM ELEMENT PURPOSE DESCRIPTOR. They are fully discussed in the description of FEM data entity FEM-17. Briefly, the FEM ELEMENT SHAPE defines the general shape of the element (e.g., line, quadrilateral, triangle, hexahedron, etc...). The FEM ELEMENT ORDER defines an element's edge parameterization (e.g., linear, cubic, etc...). The FEM ELEMENT PURPOSE DESCRIPTOR simply describes the intended purpose of the element.

Date Created: 7th April 1985 by FEM Committee

FEM/R-3 FEM Master Coordinate System

A FEM always has a (one and only one) FEM COORDINATE MASTER SYSTEM. The origin of the FEM COORDINATE MASTER SYSTEM is always at (0,0,0) and it is orthogonal.

Date Created: 7th April 1985 by FEM Committee
Date Revised: 31st March 1987 by ISO Subgroup 2

FEM/R-4 FEA

A FEA is produced by processing a FEM and a FEA CONTROL using a suitable analysis code.

Date Created: 7th April 1985 by FEM Committee
Date Revised: 31st March 1987 by ISO Subgroup 2

FEM/R-5 FEM Units

All model data must be specified in units derived from the fundamental units of measure defined by the UNITS data entity (see FEM data entity FEM-23). (Notice that the model permits a FEM, a FEA CONTROL, and a FEA to have a different set of fundamental units.)

Date Created: 16th July 1986 by FEM Committee

FEM/R-6 FEM Elements with Constant Geometric Properties

If a FEM element has a constant set of geometric properties, then only one FEM GEOMETRIC PROPERTY data entity (FEM-11) is defined for that element. This entity will be referenced from a FEM ELEMENT GEOMETRIC PROPERTY data entity (FEM-13). Stress recovery coefficients defined via the FEM ELEMENT GEOMETRIC PROPERTY imply that they are the same at each of the finite element's nodes. If a FEM element's stress recovery coefficients vary (even though no other elemental properties vary), then their values (as well as the non-varying values) must be obtained through the FEM ELEMENT/NODE GEOMETRIC PROPERTY data entity (FEM-15).
If a FEM ELEMENT GEOMETRIC PROPERTY exists for a FEM ELEMENT, then varying finite element properties are undefined.

Date Created: 16th July 1986 by FEM Committee
Date Revised: 31st March 1987 by ISO Subgroup 2

FEM/R-7 FEM Elements with Varying Geometric Properties

If a FEM element has a varying set of geometric properties, then a FEM GEOMETRIC PROPERTY data entity (FEM-11) is defined for every one of the finite element's node. These entities will be referenced via a FEM ELEMENT/NODE GEOMETRIC PROPERTY data entity (FEM-15).
Date Created: 16th July 1986 by FEM Committee
Date Revised: 31st March 1987 by ISO Subgroup 2

FEM/R-8 FEM Stress Recovery Coefficients

Finite element stress recovery coefficients are specified relative to the element local coordinate system.

Date Created: 16th July 1986 by FEM Committee

FEM/R-9 FEM Beam Shear Area

For beam finite elements, beam shear areas must be specified (not shear area factors).

Date Created: 16th July 1986 by FEM Committee

FEM/R-10 FEM Composite Laminate Z Direction

For composite laminate materials the direction of increasing Z in the local coordinate system is the direction for traversing the laminate from bottom to top through the composite layer stack.

Date Created: 22^{nd} October 1986 by FEM Committee

FEM/R-11 FEM Composite Laminate Z Offset

The Z offset of a composite laminate material is the distance to the top of the layer composite stack. Increasing FEM LAMINATE PLY SEQUENCE NUMBER indicates a top down direction through the layered composite. This means that FEM LAMINATE PLY SEQUENCE NUMBER 1 is located at the top of the laminate build (the maximum Z offset).

Date Created: 22^{nd} October 1986 by FEM Committee
Date Revised: 31^{st} March 1987 by ISO Subgroup 2

FEM/R-12 FEM Springs, Dampers, and Beams

Reference to a FEM ELEMENT/NODE GEOMETRIC PROPERTY by a FEM ELEMENT instance that is a line element such as spring, damper, 2 node beam element, etc. is undefined and therefore not allowed.

Date Created: 12^{th} October 1987 by FEM Committee

FEM/R-13 FEM Anisotropic 2D and 3D Data (obsolete)
Date Created: 12^{th} October 1987 by FEM Committee
Date Removed: 13^{th} June 1988 by FEM Committee

FEM/R-14 FEM Optional Shell Material Property Reference

The FEM SHELL GEOMETRIC PROPERTY (FEM-24) refers to three material numbers, one is for bending effects, one is for shearing effects, and one is for membrane-bending coupling effects. Specification of these material number are optional. If the material number for one of the above effects is not defined, then that effect will not be included in the shell element's stiffness matrix. (This rule is stated rather than creating three new entities with a zero or one cardinality with respect to the existing entity. Also the fact that they are missing is an important bit of data.)

Date Created: 12^{th} October 1987 by FEM Committee

FEM/R-15 Preventing Fatal Recursive Material ID References

A FEM COMPOSITE MATERIAL PROPERTY (entity FEM-35) and FEM MATERIAL TABLE (entity FEM-61) reference other materials in their make-up. That could lead to a fatal recession of the material referencing itself. In order to avoid this, no reference in the FEM COMPOSITE MATERIAL PROPERTY or FEM MATERIAL TABLE can be to itself.

Date Created: 30^{th} March 1988 by FEM Committee

9.1.4. Abbreviations and Acronyms

FEM	Finite Element Model
FEA	Finite Element Analysis

9.2 :FEM Planning Model

9.2.1 FEM Planning Model Entity Pool

The following are the data entities used in the FEM Planning model.

Entity	Entity Name (Sorted by entity number)
FEM-1	FEM
FEM-3	FEM ENVIRONMENT
FEM-6	FEM CONNECTIVITY
FEM-7	FEM GROUP
FEM-11	FEM GEOMETRIC PROPERTY
FEM-12	FEM MATERIAL PROPERTY
FEM-14	FEA RESULT
FEM-16	FEM NODE
FEM-17	FEM ELEMENT
FEM-18	FEM COORDINATE SYSTEM
FEM-21	FEA CONTROL
FEM-22	FEA
INT-1	SHAPE
PSCM-2	PRODUCT ITEM VERSION
SS-1	STANDARD SECTION
UNIT-1	UNITS

Entity	Entity Name (Sorted by entity name)
FEM-22	FEA
FEM-21	FEA CONTROL
FEM-14	FEA RESULT
FEM-1	FEM
FEM-6	FEM CONNECTIVITY
FEM-18	FEM COORDINATE SYSTEM
FEM-17	FEM ELEMENT
FEM-3	FEM ENVIRONMENT
FEM-11	FEM GEOMETRIC PROPERTY
FEM-7	FEM GROUP
FEM-12	FEM MATERIAL PROPERTY
FEM-16	FEM NODE
PSCM-2	PRODUCT ITEM VERSION
INT-1	SHAPE
SS-1	STANDARD SECTION
UNIT-1	UNITS

9.2.2 FEM Planning Model Diagram

The FEM planning model is shown in Figure 102. This IDEF1X diagram shows the major data entities and their relationships. The model depicts the following important ideas:

1. A FEM always has a relationship to a product item (PRODUCT ITEM VERSION). This relationship can be that the FEM is a mathematical model of part of a product item, a single product item, or many product items.
2. A FEM is a complete model. That is the model is complete in the sense that all of the elements, nodes, materials, groups, geometric properties are present so that a finite element analysis can be performed.
3. All of the units used within a FEM, a FEA CONTROL, and a FEA are consistent. That is to say all of the units within a FEM and a FEA are either base (fundamental) units or can be derived from the base units. Also, the base units used in a FEM, FEA CONTROL, and a FEA may be different from each other.
4. A FEA is an occurrence of a FEM and a FEA CONTROL being combined and run through a computer code. The result being a FEA with its results. A FEM and a FEA CONTROL may be independent of each other.

9.2.3 Integration Planning Model Diagram

The FEM Integration Planning Model is show in Figure 103 illustrating the five major integration points detailed below:

1. Product Item Version
   See point 1 of FEM planning model diagram.
2. Material
   Material data that will provide the necessary constants in order to perform an FEA.
3. Standard Section
   Standard Section data will provide the necessary constants in order to describe the needed beam geometric properties for FEM beam elements. For example, cross-sectional area is one of those properties needed.
4. Units
   See point 3 of FEM planning model diagram.
5. Shape
   Shape will provide all of the necessary geometric information needed to define the correct geometry for the FEM. This is independent of the method or mean of how the geometry is represented such as wire-frame, GSC, etc.

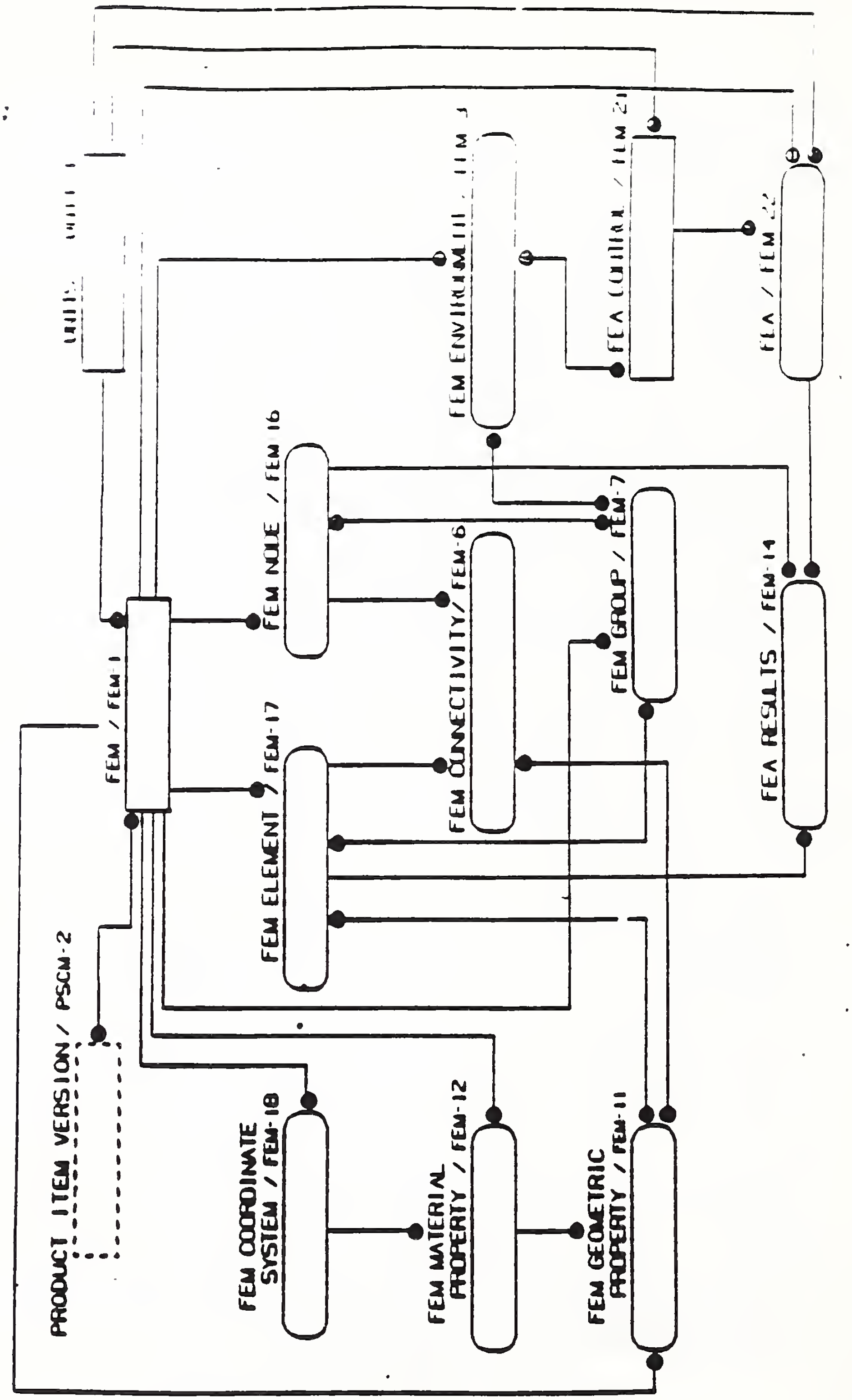


Figure D-102: FEM Planning Model Diagram

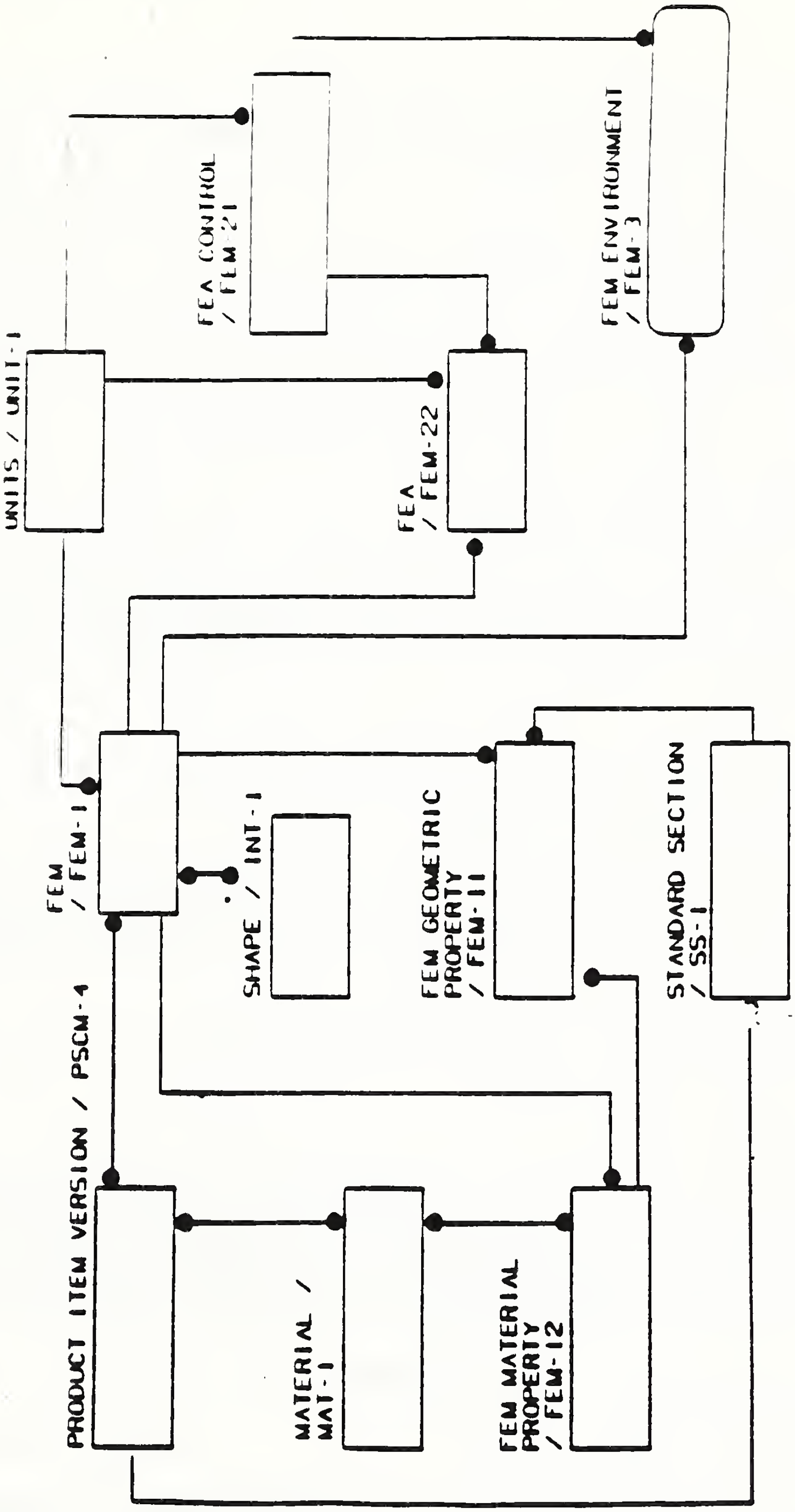


Figure D–103: FEM Integration Planning Model

9.2.4; FEM Planning Model Entity Glossary

Entity Name: **FEM**

Entity Number: **FEM-1**

A FEM is a collection of finite elements, nodes, coordinate systems, material properties, and geometric properties.

Entity Name: **PRODUCT ITEM/FEM**

Entity Number: **FEM-2**

A PRODUCT ITEM is an entity that the enterprise keeps track of; e.g., a part, a drawing, an assembly, etc... The PRODUCT ITEM/FEM intersection data entity records which PRODUCT ITEM(s) the FEM references and what FEMs have been used to describe a PRODUCT ITEM.

Entity Name: **FEM ENVIRONMENT**

Entity Number: **FEM-3**

A FEM ENVIRONMENT is any external or internal influence on a FEM. An example of an environment is the specification of loads (forces) and constraints (prescribed displacements) for a structural mechanics problem.

Entity Name: **FEM CONNECTIVITY**

Entity Number: **FEM-6**

The FEM CONNECTIVITY data entity resolves the many-to-many relationship that exists between the FEM NODE and FEM ELEMENT data entities. The attribute FEM NODE SEQUENCE NUMBER was invented to satisfy the requirements of the IDEF1X modeling methodology. It is not required to be transferred in the physical file. An ordered list of a finite element's nodes will suffice.

Entity Name: **FEM GROUP**

Entity Number: **FEM-7**

This entity contains data about a set of FEM entities that, for convenience, can be referred to by referring to one grouped item. Groups of elements and nodes care currently defined and they may be arbitrarily combined. Thus groups (sets) of nodes and finite elements can be referenced by referring to a single FEM GROUP data entity.

Entity Name: **FEM GEOMETRIC PROPERTY**

Entity Number: **FEM-11**

This entity contains data that describes the physical and geometrical characteristics of a finite element.

Entity Name: **FEM MATERIAL PROPERTY**

Entity Number: **FEM-12**

This entity contains data that describes a material constitutive matrix for either a FEM ELEMENT or a FEM NODE.

Entity Name: **FEA RESULT**

Entity Number: **FEM-14**

The FEA RESULT data entity collects data resulting from a FEA (a finite element analysis). For example, a result could be a set of stress values at a finite element's centroid.

Entity Name: **FEM NODE**

Entity Number: **FEM-16**

A FEM NODE is a place within a FEM. It is a point in space with attributes specific to a FEM.

Entity Name: **FEM ELEMENT**

Entity Number: **FEM-17**

This entity contains data about a finite element. A finite element is the basic building block of a FEM. It defines a mathematical relationship between a FEM's nodes.

Entity Name: **FEM COORDINATE SYSTEM**

Entity Number: **FEM-18**

A reference frame used to define locations of FEM elements, nodes, environments, and results in 2D or 3D space. This data entity contains information about the use (either master or definition) and type (rectangular, polar, etc...) of a coordinate system.

Entity Name: **FEA CONTROL**

Entity Number: **FEM-21**

Ancillary data pertaining to a finite element analysis. Load cases are collected into FEM ENVIRONMENTS and then applied to a FEM via a FEA CONTROL.

Entity Name: **FEA**

Entity Number: **FEM-22**

Application of a FEA CONTROL to a FEM, via a computer program, creates a finite element analysis (FEA).

Entity Name: **MATERIAL**

Entity Number: **MAT-1**

This entity contains all related data of a material.

Entity Name: **PRODUCT ITEM**

Entity Number: **PSCM-4**

A PRODUCT ITEM is an entity that the enterprise keeps track of; e.g., a part, a drawing, an assembly, etc... See PSCM document for more details.

Entity Name: **STANDARD SECTION**

Entity Number: **SS-1**

Standard Section data will provide the necessary constants in order to describe the needed beam geometric properties for FEM beam elements. For example, cross-sectional area is one of those properties needed.

Entity Name: **UNITS**

Entity Number: **UNIT-1**

The UNITS data entity defines the base (or fundamental) units of measure that a finite element model's data is expressed in.

9.3 FEM Reference Models

The FEM entity pool and reference models used to create the IDEF1X FEM global model are found in this section. Each reference model investigates a subset of FEM and FEA information.

Section 9.3.1 contains a list of the data entities used to create the FEM reference models. The actual models are presented in Section 9.3.2. The entity definitions follow in Section 9.3.3.

9.3.1 FEM Reference Model Entity Pool

The following are the entities used in the FEM reference and global data models.

Entity	Entity Name (Sorted by entity number)
FEA-1	FEA
FEAC-1	FEA CONTROL
FEM-1	FEM
FEM-2	PRODUCT ITEM/FEM
FEM-6	FEM CONNECTIVITY
FEM-7	FEM GROUP
FEM-8	FEM NODE GROUP
FEM-9	FEM ELEMENT GROUP
FEM-11	FEM GEOMETRIC PROPERTY
FEM-12	FEM MATERIAL PROPERTY
FEM-13	FEM ELEMENT GEOMETRIC PROPERTY
FEM-15	FEM ELEMENT/NODE GEOMETRIC PROPERTY
FEM-16	FEM NODE
FEM-17	FEM ELEMENT
FEM-18	FEM COORDINATE SYSTEM
FEM-19	FEM COORDINATE MASTER SYSTEM
FEM-20	FEM COORDINATE TRANSFORM
FEM-24	FEM SHELL GEOMETRIC PROPERTY
FEM-25	FEM SOLID GEOMETRIC PROPERTY
FEM-26	FEM POINT GEOMETRIC PROPERTY
FEM-27	FEM BEAM GEOMETRIC PROPERTY
FEM-28	FEM SHELL STRESS RECOVERY COEFFICIENT
FEM-29	FEM GEOMETRIC PROPERTY TEXT
FEM-30	FEM SPRING GEOMETRIC PROPERTY
FEM-31	FEM DAMPER GEOMETRIC PROPERTY
FEM-32	FEM HOMOGENEOUS MATERIAL PROPERTY
FEM-33	FEM BEAM PROPERTY DATA

Entity	Entity Name (Sorted by entity number)
FEM-34	FEM BEAM STRESS RECOVERY COEFFICIENT
FEM-35	FEM COMPOSITE MATERIAL PROPERTY
FEM-36	FEM ISOTROPIC MATERIAL PROPERTY
FEM-37	FEM ANISOTROPIC MATERIAL PROPERTY
FEM-38	FEM ISOTROPIC STRUCTURAL MATERIAL PROPERTY
FEM-39	FEM ORTHOTROPIC MATERIAL PROPERTY
FEM-40	FEM ISOTROPIC THERMAL MATERIAL PROPERTY
FEM-41	FEM ANISOTROPIC THERMAL 3D MATERIAL PROPERTY
FEM-42	FEM MIXTURE COMPOSITE MATERIAL PROPERTY
FEM-43	FEM HALPIN-TSAI MATERIAL PROPERTY
FEM-44	FEM LAMINATE COMPOSITE MATERIAL PROPERTY
FEM-45	FEM MIXTURE MATERIAL PROPERTY
FEM-46	FEM LAMINATE MATERIAL PROPERTY
FEM-47	FEM ORTHOTROPIC STRUCTURAL 2D MATERIAL PROPERTY
FEM-48	FEM ORTHOTROPIC THERMAL 2D MATERIAL PROPERTY
FEM-49	FEM ORTHOTROPIC STRUCTURAL 3D MATERIAL PROPERTY
FEM-50	FEM ORTHOTROPIC THERMAL 3D MATERIAL PROPERTY
FEM-51	FEM ANISOTROPIC STRUCTURAL 2D MATERIAL PROPERTY
FEM-52	FEM ANISOTROPIC STRUCTURAL 3D MATERIAL PROPERTY
FEM-53	FEM ANISOTROPIC THERMAL 2D MATERIAL PROPERTY
FEM-54	FEM MEMBRANE/SHELL GEOMETRIC PROPERTY
FEM-55	FEM INTEGRATION ORDER
FEM-56	FEM APPROVAL
FEM-58	FEM COORDINATE DEFINITION SYSTEM
FEM-59	FEM STRUCTURAL MASS DENSITY
FEM-60	FEM STRUCTURAL DAMPING COEFFICIENT
FEM-61	FEM MATERIAL TABLE
FEM-62	FEM MATERIAL TABLE INSTANCE
FEM-63	FEM ISOTROPIC THERMAL EXPANSION
FEM-64	FEM ORTHOTROPIC THERMAL EXPANSION
FEM-65	FEM ANISOTROPIC 2D THERMAL EXPANSION
FEM-66	FEM ANISOTROPIC 3D THERMAL EXPANSION
FEM-67	FEM LINE ELEMENT GEOMETRIC PROPERTY
FEM-68	FEM SPRING PROPERTY
FEM-69	FEM DAMPER PROPERTY
FEM-70	FEM BEAM INTERVAL
FEM-71	FEM BEAM SECTION
FEM-72	FEM BEAM STANDARD SECTION
FEM-73	FEM BEAM PIN DATA
FEM-74	FEM BEAM OFFSET VECTORS

SECTION 9: FEM INFORMATION MODEL

Entity	Entity Name (Sorted by entity number)
FEM-75	FEM BEAM WARPING DATA
FEM-76	FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY
FEM-77	FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY
FEM-78	FEM ORTHOTROPIC 2D THERMAL EXPANSION
FEM-79	FEM ORTHOTROPIC 3D THERMAL EXPANSION
FEM-80	FEM ANISOTROPIC THERMAL MATERIAL PROPERTY
FEM-81	FEM ANISOTROPIC THERMAL EXPANSION
FEM-82	FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY
PSCM-4	PRODUCT ITEM VERSION
RA-1	APPROVAL
SS-1	STANDARD SECTION
UNIT-1	UNITS

Entity	Entity Name (Sorted by entity name)
RA-1	APPROVAL
FEA-1	FEA
FEAC-1	FEA CONTROL
FEM-1	FEM
FEM-37	FEM ANISOTROPIC MATERIAL PROPERTY
FEM-82	FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY
FEM-51	FEM ANISOTROPIC STRUCTURAL 2D MATERIAL PROPERTY
FEM-52	FEM ANISOTROPIC STRUCTURAL 3D MATERIAL PROPERTY
FEM-80	FEM ANISOTROPIC THERMAL MATERIAL PROPERTY
FEM-53	FEM ANISOTROPIC THERMAL 2D MATERIAL PROPERTY
FEM-41	FEM ANISOTROPIC THERMAL 3D MATERIAL PROPERTY
FEM-81	FEM ANISOTROPIC THERMAL EXPANSION
FEM-65	FEM ANISOTROPIC 2D THERMAL EXPANSION
FEM-66	FEM ANISOTROPIC 3D THERMAL EXPANSION
FEM-56	FEM APPROVAL
FEM-27	FEM BEAM GEOMETRIC PROPERTY
FEM-70	FEM BEAM INTERVAL
FEM-74	FEM BEAM OFFSET VECTORS
FEM-73	FEM BEAM PIN DATA
FEM-33	FEM BEAM PROPERTY DATA

Entity	Entity Name (Sorted by entity name)
FEM-71	FEM BEAM SECTION
FEM-72	FEM BEAM STANDARD SECTION
FEM-75	FEM BEAM WARPING DATA
FEM-34	FEM BEAM STRESS RECOVERY COEFFICIENT
FEM-35	FEM COMPOSITE MATERIAL PROPERTY
FEM-6	FEM CONNECTIVITY
FEM-58	FEM COORDINATE DEFINITION SYSTEM
FEM-19	FEM COORDINATE MASTER SYSTEM
FEM-18	FEM COORDINATE SYSTEM
FEM-20	FEM COORDINATE TRANSFORM
FEM-31	FEM DAMPER GEOMETRIC PROPERTY
FEM-69	FEM DAMPER PROPERTY
FEM-17	FEM ELEMENT
FEM-13	FEM ELEMENT GEOMETRIC PROPERTY
FEM-9	FEM ELEMENT GROUP
FEM-15	FEM ELEMENT/NODE GEOMETRIC PROPERTY
FEM-11	FEM GEOMETRIC PROPERTY
FEM-29	FEM GEOMETRIC PROPERTY TEXT
FEM-7	FEM GROUP
FEM-43	FEM HALPIN-TSAI MATERIAL PROPERTY
FEM-55	FEM INTEGRATION ORDER
FEM-36	FEM ISOTROPIC MATERIAL PROPERTY
FEM-38	FEM ISOTROPIC STRUCTURAL MATERIAL PROPERTY
FEM-63	FEM ISOTROPIC THERMAL EXPANSION
FEM-40	FEM ISOTROPIC THERMAL MATERIAL PROPERTY
FEM-32	FEM HOMOGENEOUS MATERIAL PROPERTY
FEM-44	FEM LAMINATE COMPOSITE MATERIAL PROPERTY
FEM-46	FEM LAMINATE MATERIAL PROPERTY
FEM-67	FEM LINE ELEMENT GEOMETRIC PROPERTY
FEM-12	FEM MATERIAL PROPERTY
FEM-61	FEM MATERIAL TABLE
FEM-62	FEM MATERIAL TABLE INSTANCE
FEM-54	FEM MEMBRANE/SHELL GEOMETRIC PROPERTY
FEM-45	FEM MIXTURE MATERIAL PROPERTY
FEM-42	FEM MIXTURE COMPOSITE MATERIAL PROPERTY
FEM-16	FEM NODE
FEM-8	FEM NODE GROUP
FEM-39	FEM ORTHOTROPIC MATERIAL PROPERTY
FEM-76	FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY
FEM-47	FEM ORTHOTROPIC STRUCTURAL 2D MATERIAL PROPERTY
FEM-49	FEM ORTHOTROPIC STRUCTURAL 3D MATERIAL PROPERTY

Entity	Entity Name (Sorted by entity name)
FEM-64	FEM ORTHOTROPIC THERMAL EXPANSION
FEM-77	FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY
FEM-48	FEM ORTHOTROPIC THERMAL 2D MATERIAL PROPERTY
FEM-50	FEM ORTHOTROPIC THERMAL 3D MATERIAL PROPERTY
FEM-78	FEM ORTHOTROPIC 2D THERMAL EXPANSION
FEM-79	FEM ORTHOTROPIC 3D THERMAL EXPANSION
FEM-26	FEM POINT GEOMETRIC PROPERTY
FEM-24	FEM SHELL GEOMETRIC PROPERTY
FEM-28	FEM SHELL STRESS RECOVERY COEFFICIENT
FEM-25	FEM SOLID GEOMETRIC PROPERTY
FEM-30	FEM SPRING GEOMETRIC PROPERTY
FEM-68	FEM SPRING PROPERTY
FEM-60	FEM STRUCTURAL DAMPING COEFFICIENT
FEM-59	FEM STRUCTURAL MASS DENSITY
PSCM-4	PRODUCT ITEM VERSION
FEM-2	PRODUCT ITEM/FEM
SS-1	STANDARD SECTION
UNIT-1	UNITS

9.3.2 FEM Reference Models

This section contains all of the individual FEM reference models used to create the FEM complete IDEF1X data model. The format followed is before each reference model is presented, there is explanatory text that may apply to more than one diagram.

The following list tabulates the name and figure number of each FEM reference model.

Figure Number	Reference Model Name
104	FEM, FEA Control, and FEA
105	FEM Elements and Nodes
106	FEM Coordinate Systems
107	FEM Group
108	FEM Material and Geometric Properties
109	FEM Geometric Properties
110	FEM Point Geometric Property
111	FEM Line Geometric Properties
112	FEM Beam Geometric Property
113	FEM Beam Cross_section Data
114	FEM Shell Geometric Property
115	FEM Solid Geometric Property
116	FEM Material Properties
117	FEM Composite Material Property
118	FEM Material Table
119	FEM Homogeneous Material Property
120	FEM Isotropic Material Property
121	FEM Orthotropic Material Property
122	FEM Orthotropic Structural Material Property
123	FEM Orthotropic Thermal Material Property
124	FEM Orthotropic Thermal Expansion Material Property
125	FEM Anisotropic Material Property
126	FEM Anisotropic Structural Material Property
127	FEM Anisotropic Thermal Material Property
128	FEM Anisotropic Thermal Expansion Material Property

FEM, FEA Control, and FEA

Figure 104 depicts how a FEM and a FEA CONTROL are combined to produce a FEA. The relationship with PRODUCT ITEM (ie. the item the analyst is analysising) is shown. Also introduced is the concept that Finite Element Model may be approved or signed off through the entity FEM APPROVAL. UNITS are related to a FEA, FEA CONTROL, and FEA.

FEM rules FEM/R-1, 4, and 5 apply to this IDEF1X diagram.

FEM ELEMENT AND NODES

Figure 105 shows the heart of a FEM. A FEM will have many FEM ELEMENTs and FEM NODEs. A many-to-many relationship exists between FEM ELEMENTs and FEM NODEs; i.e., a FEM ELEMENT references many FEM NODEs and a FEM NODE may be a member

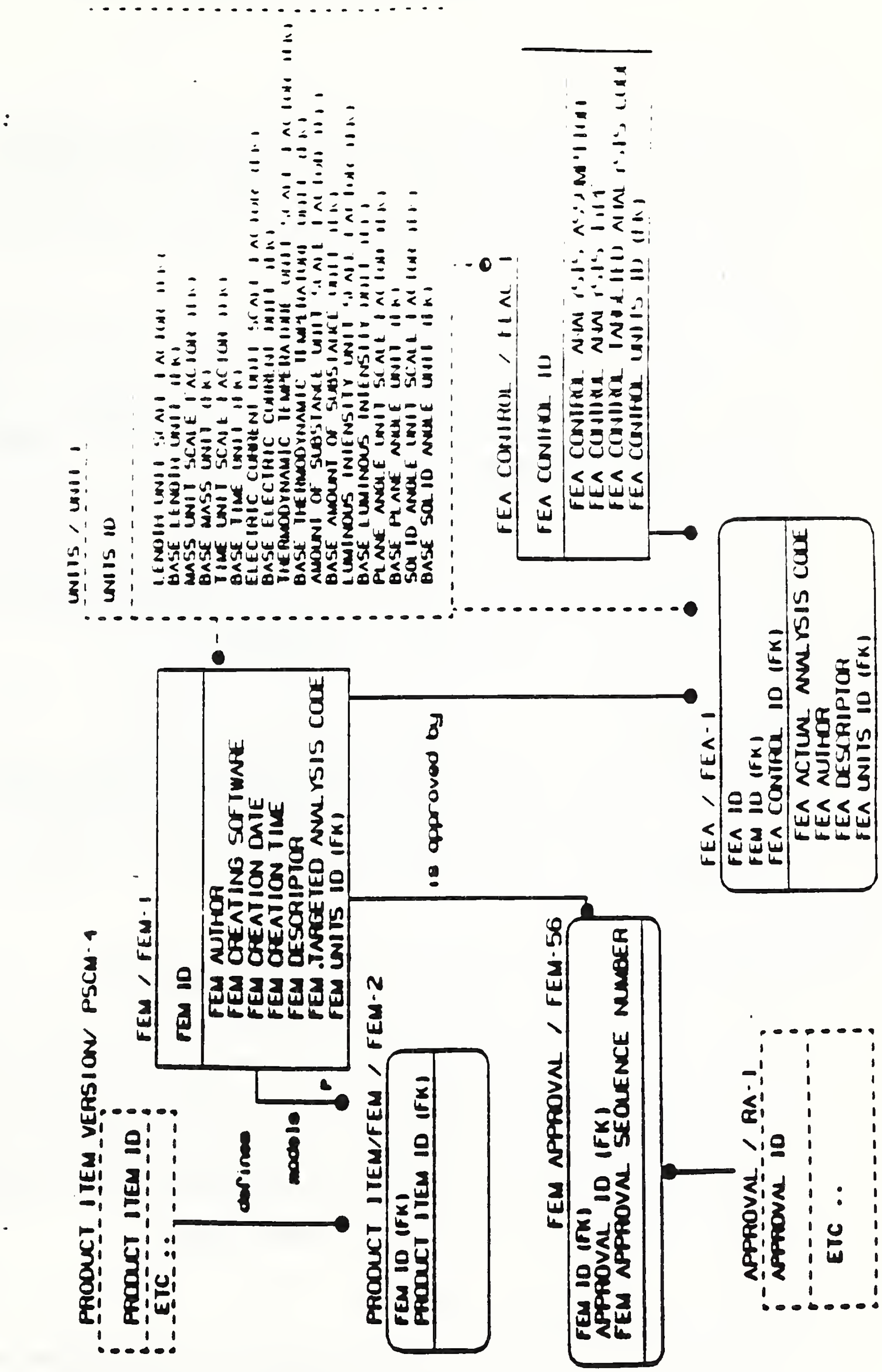

UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)
FEA CONTROL / FEAC-1
FEA CONTROL ID
FEA CONTROL ANALYSIS ASSUMPTION
FEA CONTROL ANALYSIS TYPE
FEA CONTROL TARGETED ANALYSIS CODE
FEA CONTROL UNITS ID (FK)
PRODUCT ITEM VERSION/ PSCM-4
PRODUCT ITEM ID
ETC ..
defines
models
FEM / FEM-1
FEM ID
FEM AUTHOR
FEM CREATING SOFTWARE
FEM CREATION DATE
FEM CREATION TIME
FEM DESCRIPTOR
FEM TARGETED ANALYSIS CODE
FEM UNITS ID (FK)
P
is approved by
PRODUCT ITEM/FEM / FEM-2
FEM ID (FK)
PRODUCT ITEM ID (FK)
FEA / FEA-1
FEA ID
FEM ID (FK)
FEA CONTROL ID (FK)
FEA ACTUAL ANALYSIS CODE
FEA AUTHOR
FEA DESCRIPTOR
FEA UNITS ID (FK)
FEM APPROVAL / FEM-56
FEM ID (FK)
APPROVAL ID (FK)
FEM APPROVAL SEQUENCE NUMBER
APPROVAL / RA-1
APPROVAL ID
ETC ..

Figure D-104: FEM, FEA Control, and FEA

of many FEM ELEMENTs. This many-to-many relationship is resolved by the FEM CONNECTIVITY data entity and the concept of a FEM NODE SEQUENCE NUMBER. This construct is an artifact of the IDEF1X method. Data transfer of FEM NODE SEQUENCE NUMBERs should be done in an implicit manner via an ordered list of the FEM element's nodes. The FEM NODE SEQUENCE NUMBER should not exist outside of the STEP logical and application schemas.

FEM rule FEM/R-2 applies to this IDEF1X diagram.

FEM COORDINATE SYSTEM

The relationship between FEM COORDINATE SYSTEM and FEM, FEM ELEMENT, FEM NODE, and FEM ENVIRONMENT is presented in Figure 106. The FEM COORDINATE SYSTEM definition permits reference to either a FEM COORDINATE MASTER SYSTEM or a coordinate system that directly references the FEM COORDINATE MASTER SYSTEM. (This definition is similar to the SHAPE/SIZE COORDINATE SYSTEM data entity that has been shown on a version of the IDEF1X SHAPE/SIZE global reference model.)

FEM rule FEM/R-3 applies to this IDEF1X diagram.

FEM GROUP

FEM ELEMENTs and FEM NODEs can be members of FEM ELEMENT GROUPs and FEM NODE GROUPs, respectively. A FEM GROUP may contain any number of FEM ELEMENT GROUPs and FEM NODE GROUPs. Thus, as shown in Figure 107 an arbitrary set of FEM ELEMENTs and FEM NODEs can be collected and referenced as a FEM GROUP.

FEM MATERIAL AND GEOMETRIC PROPERTIES

A FEM must have FEM MATERIAL PROPERTY(s) and FEM GEOMETRIC PROPERTY(s), as shown in Figure 108. If a finite element's geometric properties are constant over its domain, then the finite element references a FEM GEOMETRIC PROPERTY, else the FEM ELEMENT/NODE GEOMETRIC PROPERTY is referenced. (Again, the concept of FEM NODE SEQUENCE NUMBER is an artifact of the IDEF1X modeling process and should not be translated into the physical file. A simple ordered list will suffice.)

FEM rules FEM/R-6 and 7 apply to this IDEF1X diagram.

FEM GEOMETRIC PROPERTIES

The five subtypes of FEM Geometric Properties are illustrated in figure 109. Also presented in figure 109 are the relationship of the FEM INTEGRATION ORDER and FEM GEOMETRY TEXT entities to FEM GEOMETRIC PROPERTY. The combination of the data in FEM

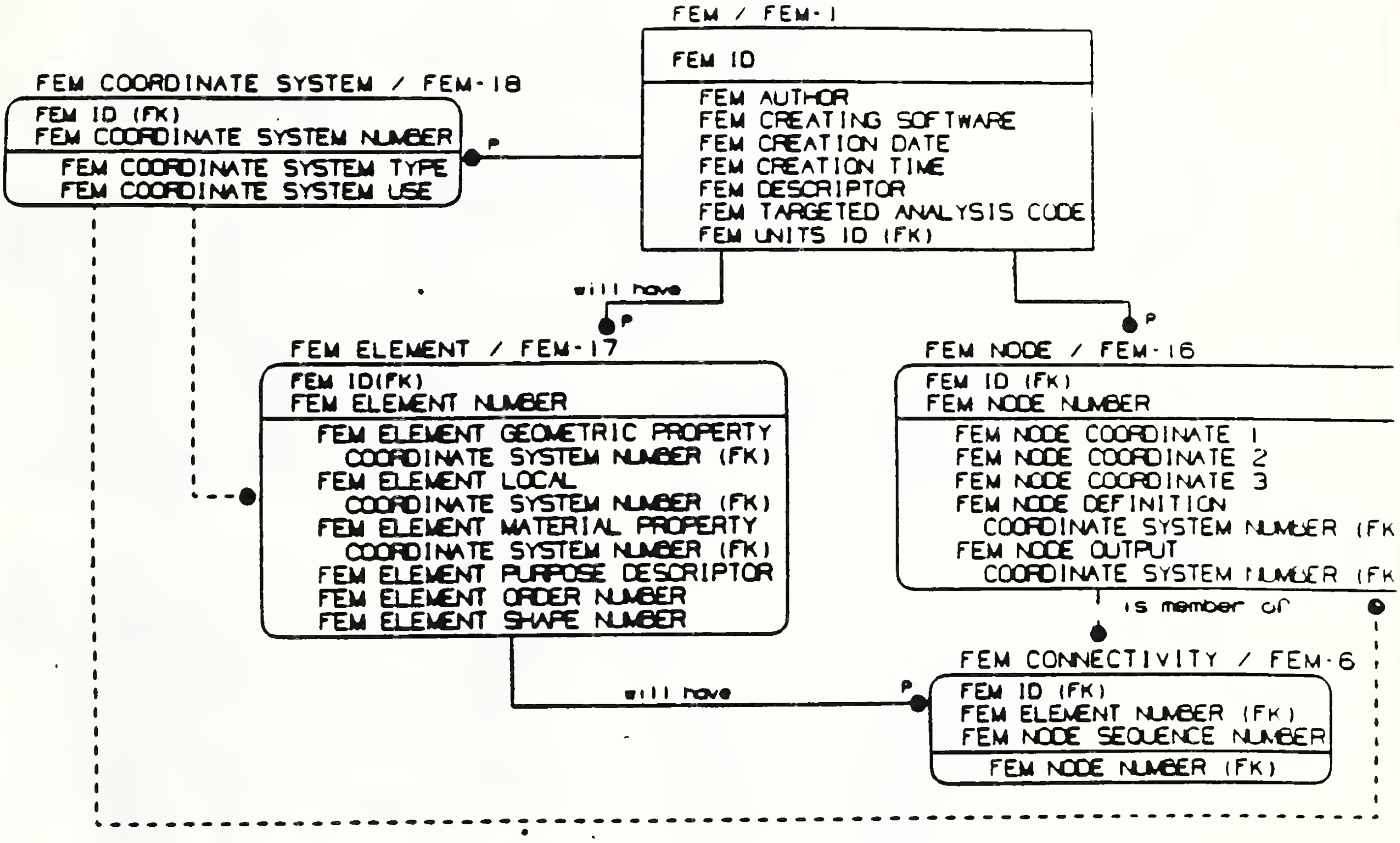


Figure D-105: FEM Elements and Nodes

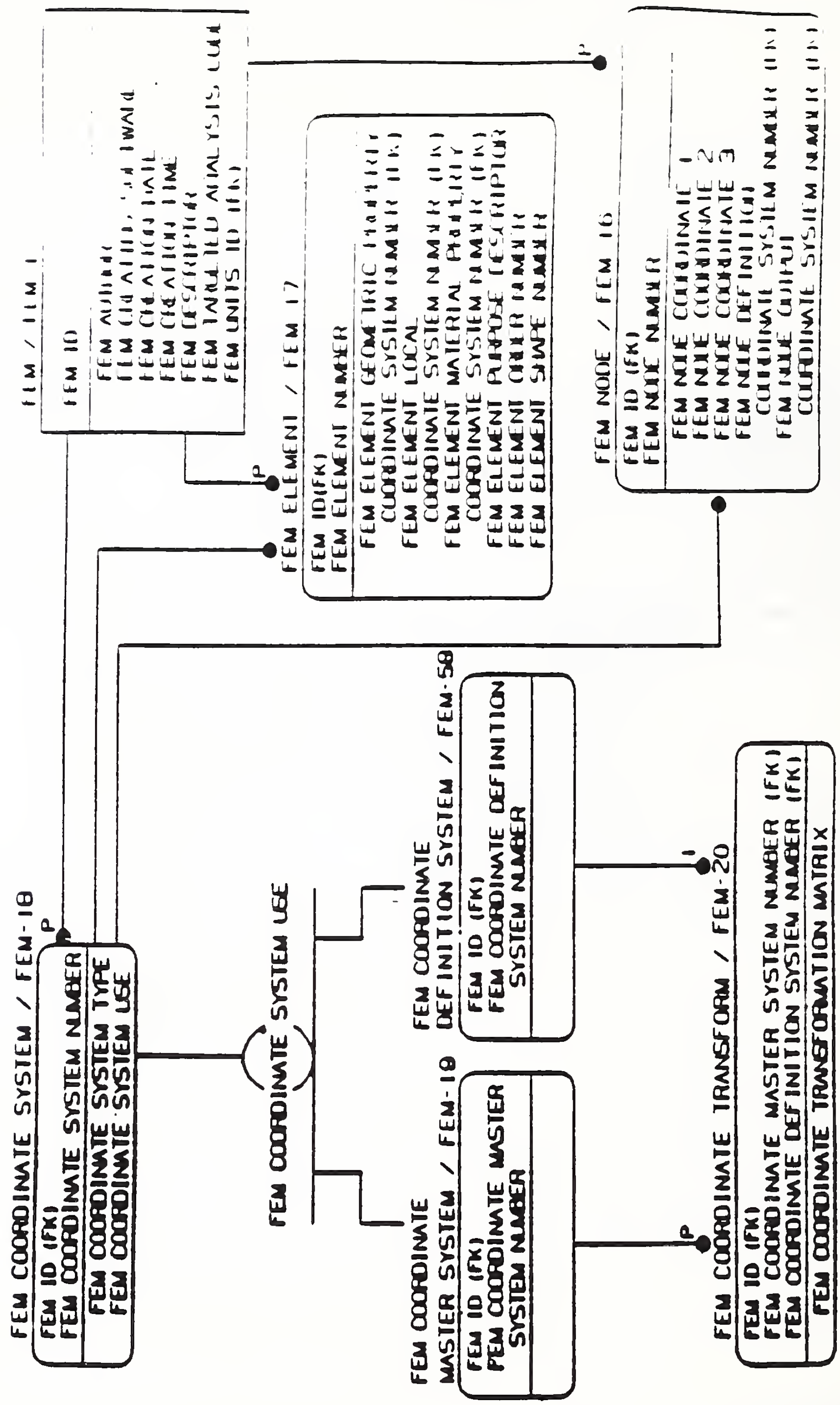

FEM / FEM 1
FEM ID
FEM AUTHOR
FEM CREATION SOFTWARE
FEM CREATION DATE
FEM CREATION TIME
FEM DESCRIPTOR
FEM TABULATED ANALYSIS CODE
FEM UNITS ID (FK)
P
FEM ELEMENT / FEM 17
FEM ID(FK)
FEM ELEMENT NUMBER
FEM ELEMENT GEOMETRIC PROPERTY COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT LOCAL COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT MATERIAL PROPERTY COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT PURPOSE DESCRIPTOR
FEM ELEMENT ORDER NUMBER
FEM ELEMENT SHAPE NUMBER
P
FEM NODE / FEM 16
FEM ID (FK)
FEM NODE NUMBER
FEM NODE COORDINATE 1
FEM NODE COORDINATE 2
FEM NODE COORDINATE 3
FEM NODE DEFINITION COORDINATE SYSTEM NUMBER (FK)
FEM NODE OUTPUT COORDINATE SYSTEM NUMBER (FK)
P
FEM COORDINATE SYSTEM / FEM-18
FEM ID (FK)
FEM COORDINATE SYSTEM NUMBER
FEM COORDINATE SYSTEM TYPE
FEM COORDINATE SYSTEM USE
FEM COORDINATE SYSTEM USE
FEM COORDINATE MASTER SYSTEM / FEM-19
FEM ID (FK)
FEM COORDINATE MASTER SYSTEM NUMBER
FEM COORDINATE DEFINITION SYSTEM / FEM-58
FEM ID (FK)
FEM COORDINATE DEFINITION SYSTEM NUMBER
P
1
FEM COORDINATE TRANSFORM / FEM-20
FEM ID (FK)
FEM COORDINATE MASTER SYSTEM NUMBER (FK)
FEM COORDINATE DEFINITION SYSTEM NUMBER (FK)
FEM COORDINATE TRANSFORMATION MATRIX

Figure D-106: FEM Coordinate Systems

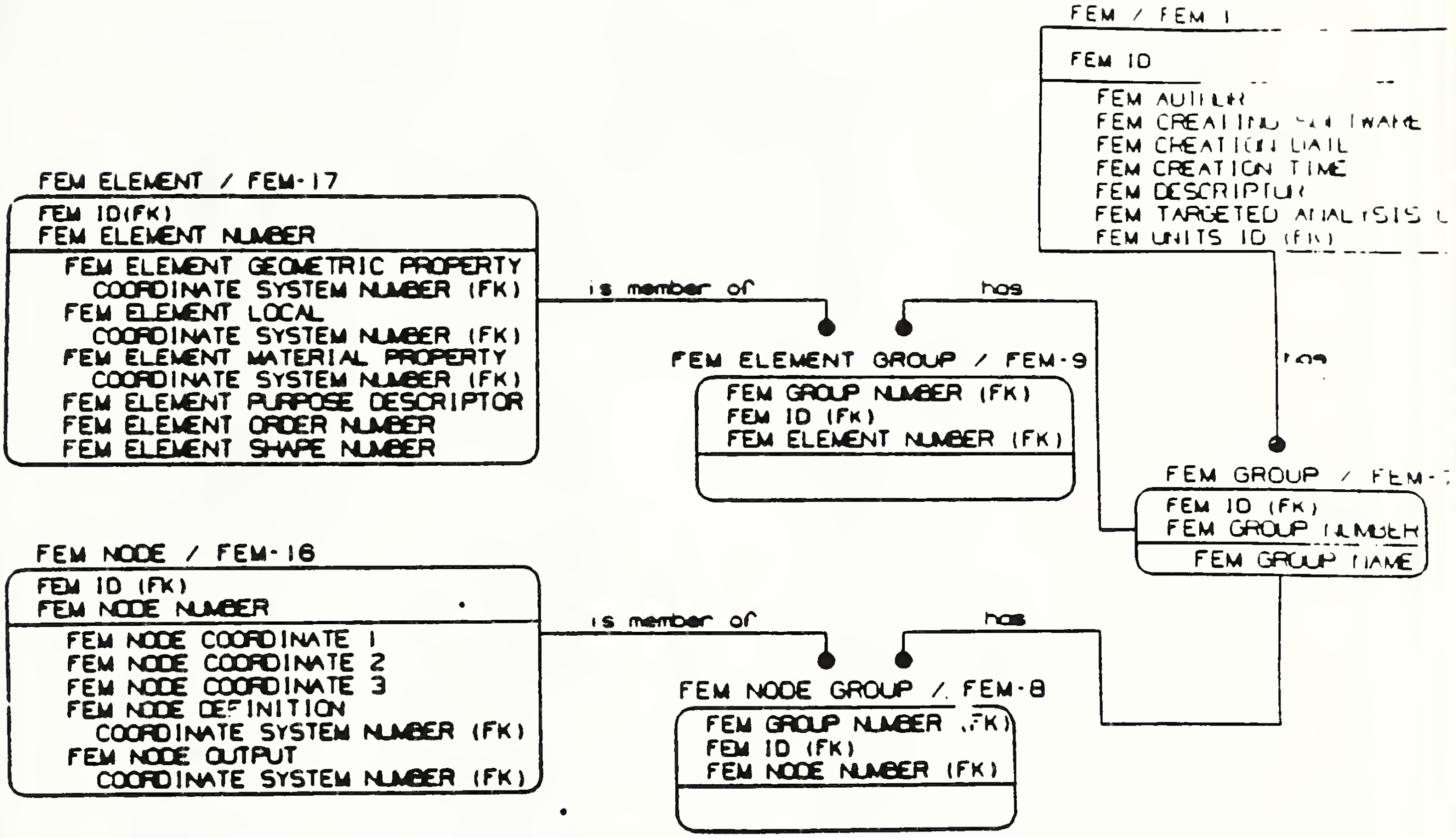


Figure D-107: FEM Group

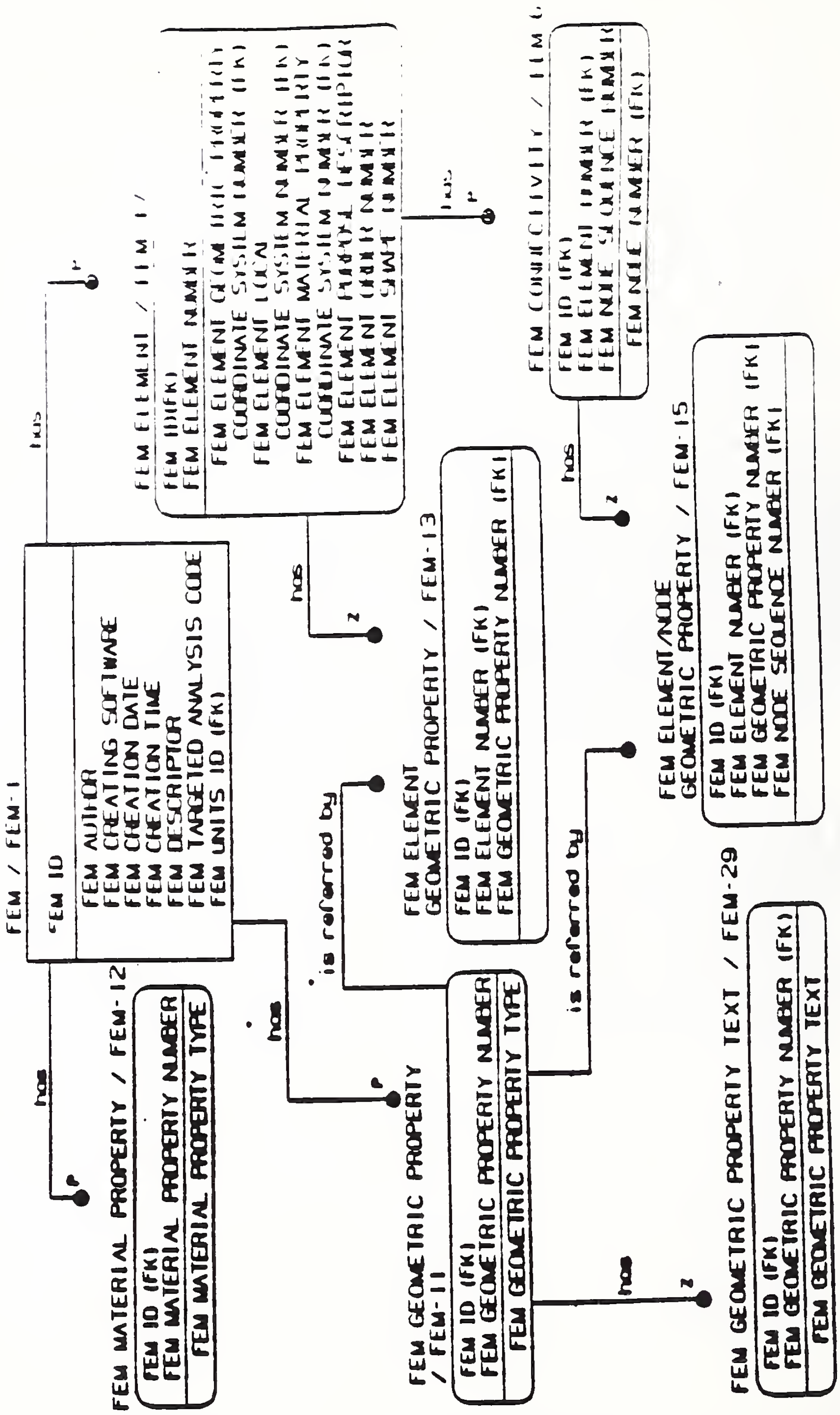

FEM / FEM-1
FEM ID
FEM AUTHOR
FEM CREATING SOFTWARE
FEM CREATION DATE
FEM CREATION TIME
FEM DESCRIPTOR
FEM TARGETED ANALYSIS CODE
FEM UNITS ID (FK)
FEM ELEMENT / FEM-7
FEM ID (FK)
FEM ELEMENT NUMBER
FEM ELEMENT GEOMETRIC PROPERTY
COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT LOCAL
COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT MATERIAL PROPERTY
COORDINATE SYSTEM NUMBER (FK)
FEM ELEMENT PURPOSE DESCRIPTOR
FEM ELEMENT ORDER NUMBER
FEM ELEMENT SHAPE NUMBER
FEM CONNECTIVITY / FEM-6
FEM ID (FK)
FEM ELEMENT NUMBER (FK)
FEM NODE SEQUENCE NUMBER
FEM NODE NUMBER (FK)
FEM MATERIAL PROPERTY / FEM-12
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM ELEMENT GEOMETRIC PROPERTY / FEM-13
FEM ID (FK)
FEM ELEMENT NUMBER (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM ELEMENT/NODE GEOMETRIC PROPERTY / FEM-15
FEM ID (FK)
FEM ELEMENT NUMBER (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM NODE SEQUENCE NUMBER (FK)
FEM GEOMETRIC PROPERTY TEXT / FEM-29
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM GEOMETRIC PROPERTY TEXT
has
P
Z
is referred by

Figure D-108: FEM Material and Geometric Properties

GEOMETRIC PROPERTY type is intended to uniquely identify all of the geometric property and material property data needed in order to formulate an element.

FEM GEOMETRIC PROPERTIES - CONTINUED

The next six figures present the six major sub-types of FEM GEOMETRIC PROPERTY information. These are:

1. FEM POINT GEOMETRIC PROPERTY, Figure 110;
2. FEM LINE GEOMETRIC PROPERTY, Figure 111;
3. FEM BEAM GEOMETRIC PROPERTY, Figure 112;
4. FEM BEAM CROSS_SECTION DATA, Figure 113;
5. FEM SHELL GEOMETRIC PROPERTY, Figure 114;
6. FEM SOLID GEOMETRIC PROPERTY, Figure 115;

FEM rules FEM/R-8, 9, 12 and 14 apply to these IDEF1X diagrams.

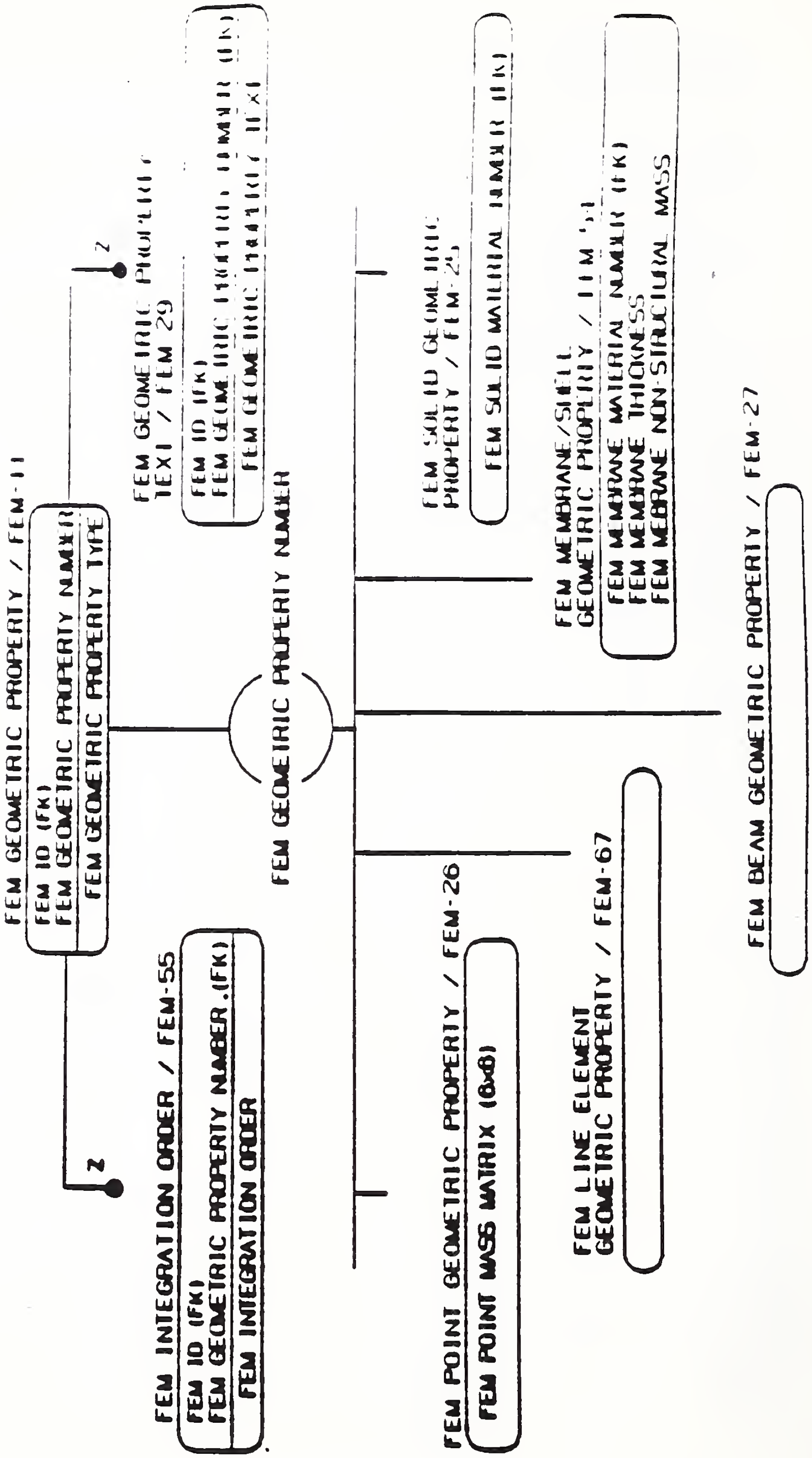

FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
Z
FEM GEOMETRIC PROPERTY TEXT / FEM 29
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM GEOMETRIC PROPERTY TEXT
Z
FEM INTEGRATION ORDER / FEM-55
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM INTEGRATION ORDER
FEM GEOMETRIC PROPERTY NUMBER
FEM SOLID GEOMETRIC PROPERTY / FEM-25
FEM SOLID MATERIAL NUMBER (FK)
FEM MEMBRANE/SHELL GEOMETRIC PROPERTY / FEM
FEM MEMBRANE MATERIAL NUMBER (FK)
FEM MEMBRANE THICKNESS
FEM MEBRANE NON-STRUCTURAL MASS
FEM BEAM GEOMETRIC PROPERTY / FEM-27
FEM POINT GEOMETRIC PROPERTY / FEM-26
FEM POINT MASS MATRIX (6x6)
FEM LINE ELEMENT GEOMETRIC PROPERTY / FEM-67

Figure D-109: FEM Geometric Properties

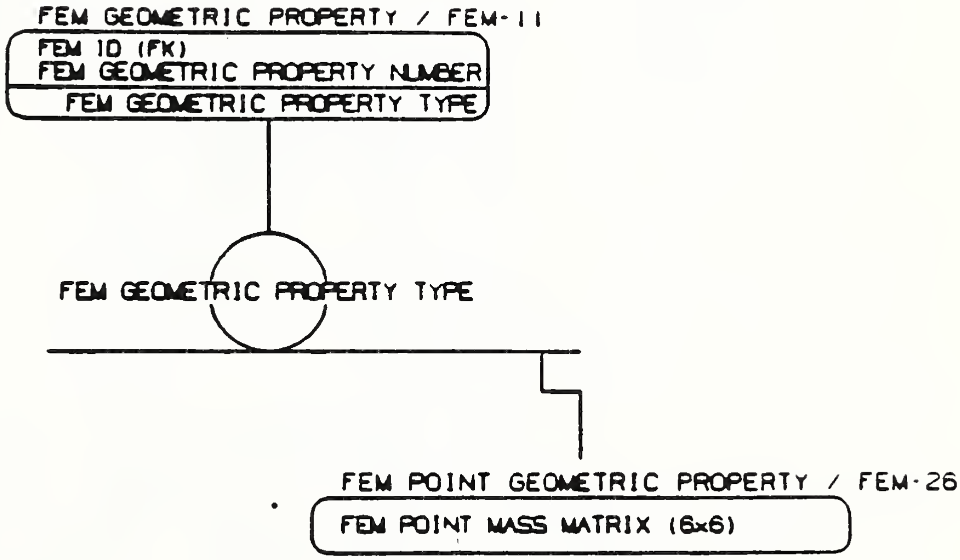

FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM GEOMETRIC PROPERTY TYPE
FEM POINT GEOMETRIC PROPERTY / FEM-26
FEM POINT MASS MATRIX (6x6)

Figure D-110: FEM Point Geometric Property

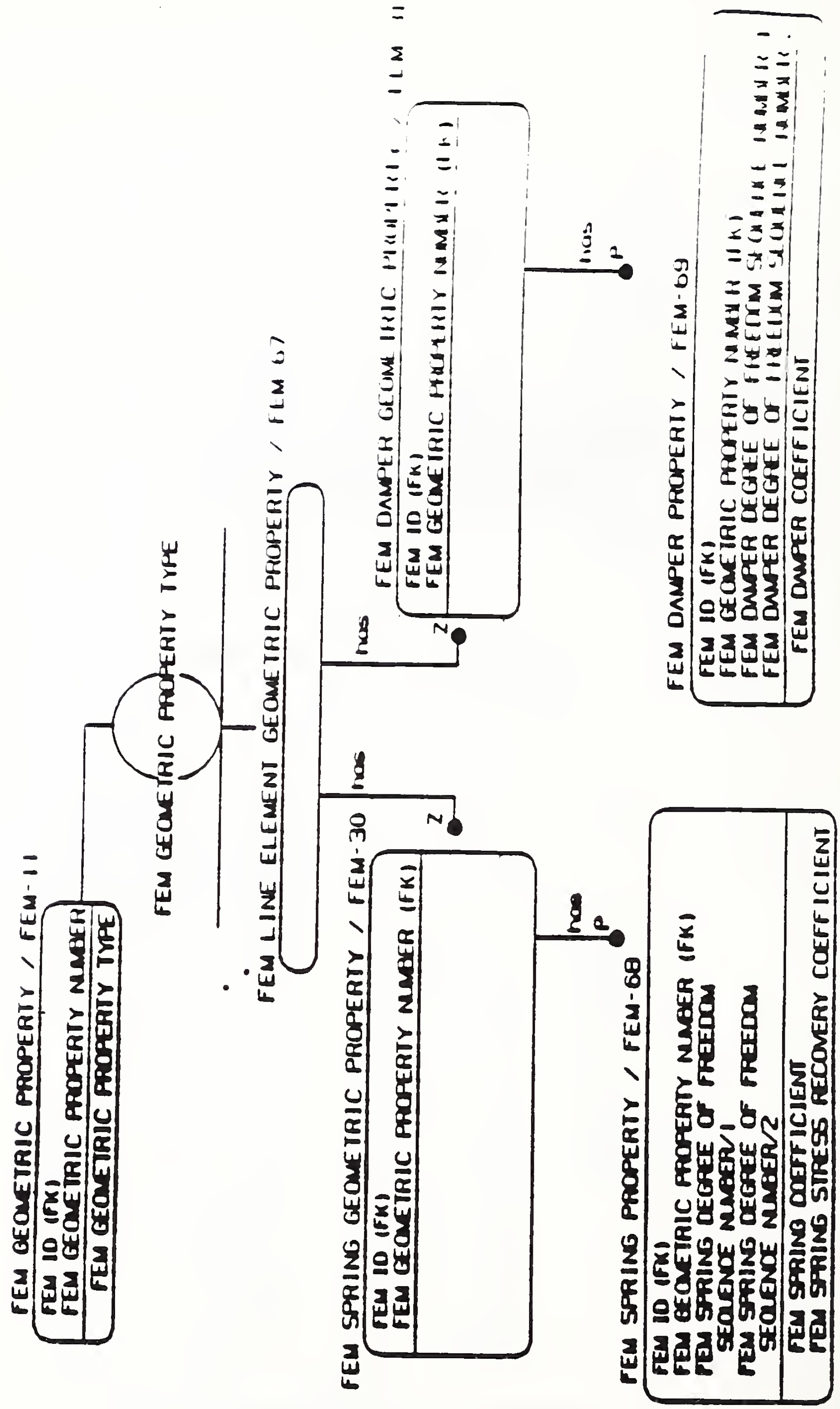

FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM GEOMETRIC PROPERTY TYPE
FEM LINE ELEMENT GEOMETRIC PROPERTY / FEM-67
has
Z
has
Z
FEM SPRING GEOMETRIC PROPERTY / FEM-30
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
has
P
FEM SPRING PROPERTY / FEM-68
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM SPRING DEGREE OF FREEDOM SEQUENCE NUMBER/1
FEM SPRING DEGREE OF FREEDOM SEQUENCE NUMBER/2
FEM SPRING COEFFICIENT
FEM SPRING STRESS RECOVERY COEFFICIENT
FEM DAMPER GEOMETRIC PROPERTY
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
has
P
FEM DAMPER PROPERTY / FEM-69
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM DAMPER DEGREE OF FREEDOM
FEM DAMPER DEGREE OF FREEDOM
FEM DAMPER COEFFICIENT

Figure D-111: FEM Line Geometric Property

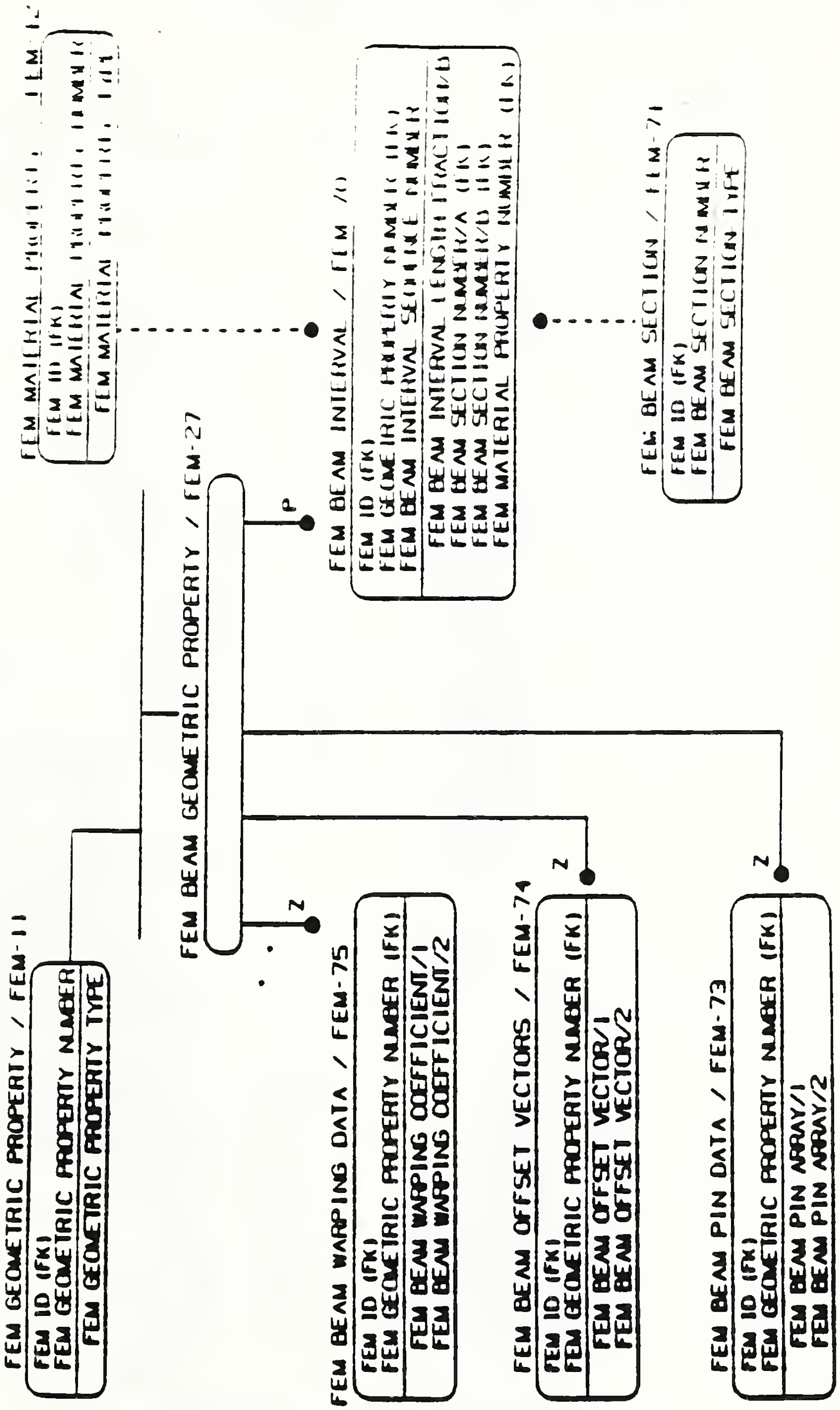

FEM MATERIAL PROPERTY / FEM-1
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
FEM BEAM INTERVAL / FEM-70
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM BEAM INTERVAL SEQUENCE NUMBER
FEM BEAM INTERVAL LENGTH FRACTION/D
FEM BEAM SECTION NUMBER/A (FK)
FEM BEAM SECTION NUMBER/B (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM BEAM SECTION / FEM-71
FEM ID (FK)
FEM BEAM SECTION NUMBER
FEM BEAM SECTION TYPE
FEM BEAM GEOMETRIC PROPERTY / FEM-27
P
Z
Z
Z
FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM BEAM WARPING DATA / FEM-75
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM BEAM WARPING COEFFICIENT/1
FEM BEAM WARPING COEFFICIENT/2
FEM BEAM OFFSET VECTORS / FEM-74
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM BEAM OFFSET VECTOR/1
FEM BEAM OFFSET VECTOR/2
FEM BEAM PIN DATA / FEM-73
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM BEAM PIN ARRAY/1
FEM BEAM PIN ARRAY/2

Figure D-112: FEM Beam Geometric Property

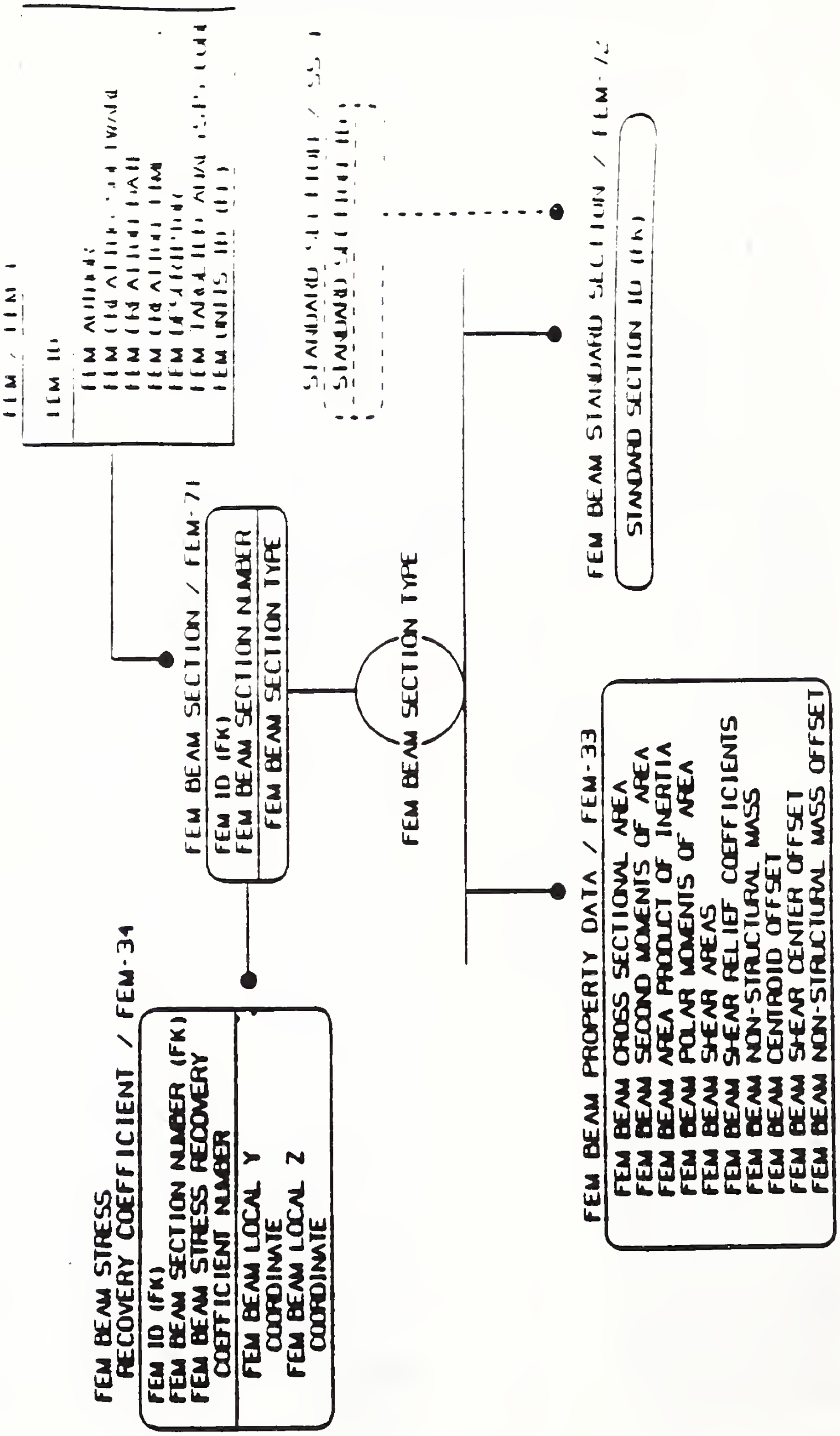

FEM / FEM-1
FEM ID
FEM AUTHOR
STANDARD SECTION / SS-1
STANDARD SECTION ID
FEM BEAM STANDARD SECTION / FEM-72
STANDARD SECTION ID (FK)
FEM BEAM SECTION / FEM-71
FEM ID (FK)
FEM BEAM SECTION NUMBER
FEM BEAM SECTION TYPE
FEM BEAM SECTION TYPE
FEM BEAM PROPERTY DATA / FEM-33
FEM BEAM CROSS SECTIONAL AREA
FEM BEAM SECOND MOMENTS OF AREA
FEM BEAM AREA PRODUCT OF INERTIA
FEM BEAM POLAR MOMENTS OF AREA
FEM BEAM SHEAR AREAS
FEM BEAM SHEAR RELIEF COEFFICIENTS
FEM BEAM NON-STRUCTURAL MASS
FEM BEAM CENTROID OFFSET
FEM BEAM SHEAR CENTER OFFSET
FEM BEAM NON-STRUCTURAL MASS OFFSET
FEM BEAM STRESS RECOVERY COEFFICIENT / FEM-34
FEM ID (FK)
FEM BEAM SECTION NUMBER (FK)
FEM BEAM STRESS RECOVERY COEFFICIENT NUMBER
FEM BEAM LOCAL Y COORDINATE
FEM BEAM LOCAL Z COORDINATE

Figure D-113: FEM Beam Cross-Section Data

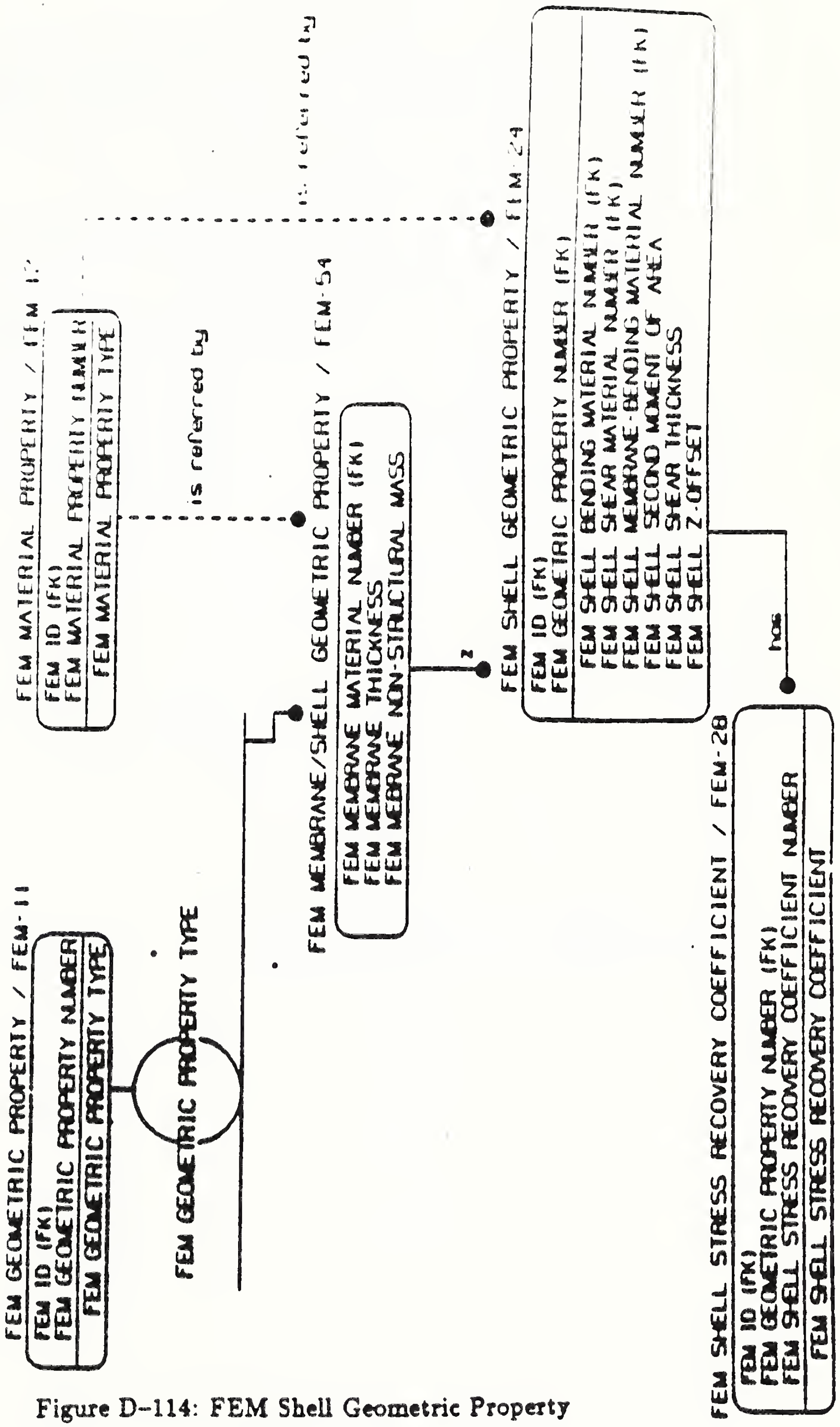

FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM GEOMETRIC PROPERTY TYPE
FEM MATERIAL PROPERTY / FEM-1?
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
is referred by
is referred by
FEM MEMBRANE/SHELL GEOMETRIC PROPERTY / FEM-54
FEM MEMBRANE MATERIAL NUMBER (FK)
FEM MEMBRANE THICKNESS
FEM MEBRANE NON-STRUCTURAL MASS
Z
FEM SHELL GEOMETRIC PROPERTY / FEM-24
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM SHELL BENDING MATERIAL NUMBER (FK)
FEM SHELL SHEAR MATERIAL NUMBER (FK)
FEM SHELL MEMBRANE-BENDING MATERIAL NUMBER (FK)
FEM SHELL SECOND MOMENT OF AREA
FEM SHELL SHEAR THICKNESS
FEM SHELL Z-OFFSET
has
FEM SHELL STRESS RECOVERY COEFFICIENT / FEM-28
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER (FK)
FEM SHELL STRESS RECOVERY COEFFICIENT NUMBER
FEM SHELL STRESS RECOVERY COEFFICIENT

Figure D-114: FEM Shell Geometric Property

FEM MATERIAL PROPERTIES

An overall view of FEM material properties is presented in figure 116. That figure (116) displays the three main super types of the materials. They are FEM COMPOSITE MATERIAL, FEM MATERIAL TABLE, and FEM HOMOGENEOUS MATERIAL PROPERTY. These are detailed in the following figures:

1. FEM MATERIAL PROPERTIES, figure 116;
2. FEM COMPOSITE MATERIAL PROPERTY, figure 117;
3. FEM MATERIAL TABLE, figure 118;
4. FEM HOMOGENEOUS MATERIAL PROPERTY, figure 119;

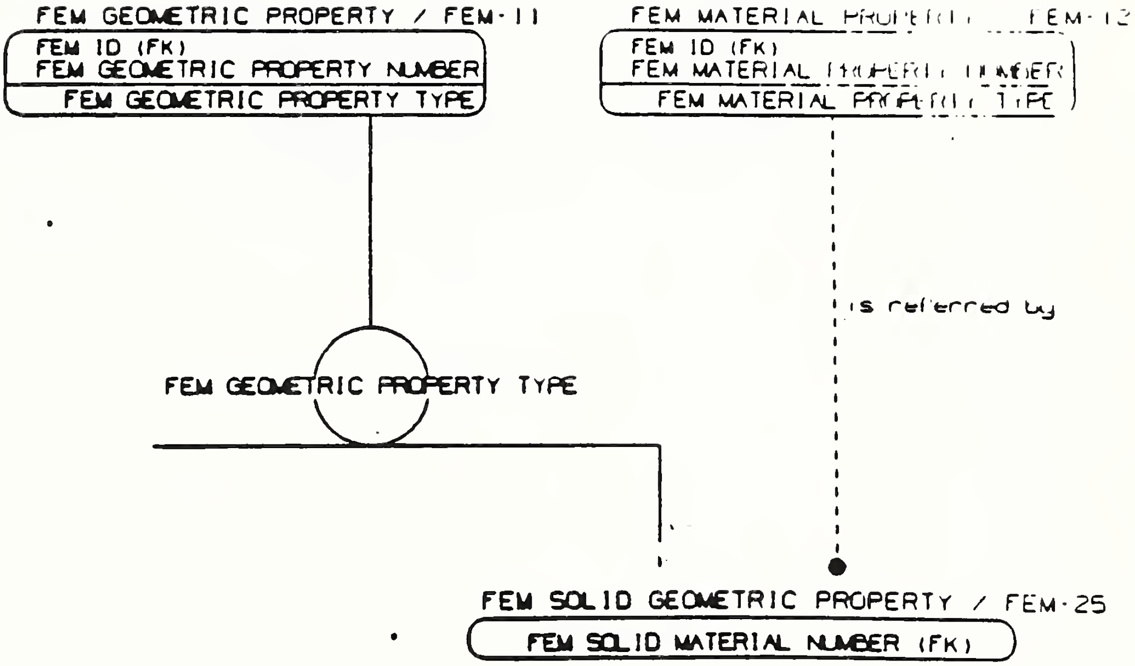

FEM GEOMETRIC PROPERTY / FEM-11
FEM ID (FK)
FEM GEOMETRIC PROPERTY NUMBER
FEM GEOMETRIC PROPERTY TYPE
FEM MATERIAL PROPERTY / FEM-12
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
is referred by
FEM GEOMETRIC PROPERTY TYPE
FEM SOLID GEOMETRIC PROPERTY / FEM-25
FEM SOLID MATERIAL NUMBER (FK)

Figure D-115: FEM Solid Geometric Property

FEM COMPOSITE MATERIAL PROPERTY

The FEM COMPOSITE MATERIAL PROPERTY is show in figure 117. That figure (117) displays three types of composite material that the FEM committee has currently defined. (It is expected that more types of composite materials would be added at later date). They are:

1. FEM MIXTURE MATERIAL PROPERTY
   This is a self contained material definition for a composite material using the theory of mixtures in order to create the composite material properties.
2. FEM HALPIN-TSAI MATERIAL PROPERTY
   This is a self contained material definition for a composite material using the theory of Halpin-Tsai in order to create the composite material properties.
3. FEM LAMINATE COMPOSITE MATERIAL PROPERTY
   This is an ordered list of FEM LAMINATE MATERIAL Properties that are stacked one upon another that creates a composite material.

Rules FEM/R-10, FEM/R-11 and FEM/R-15 apply to this IDEF1X diagram.

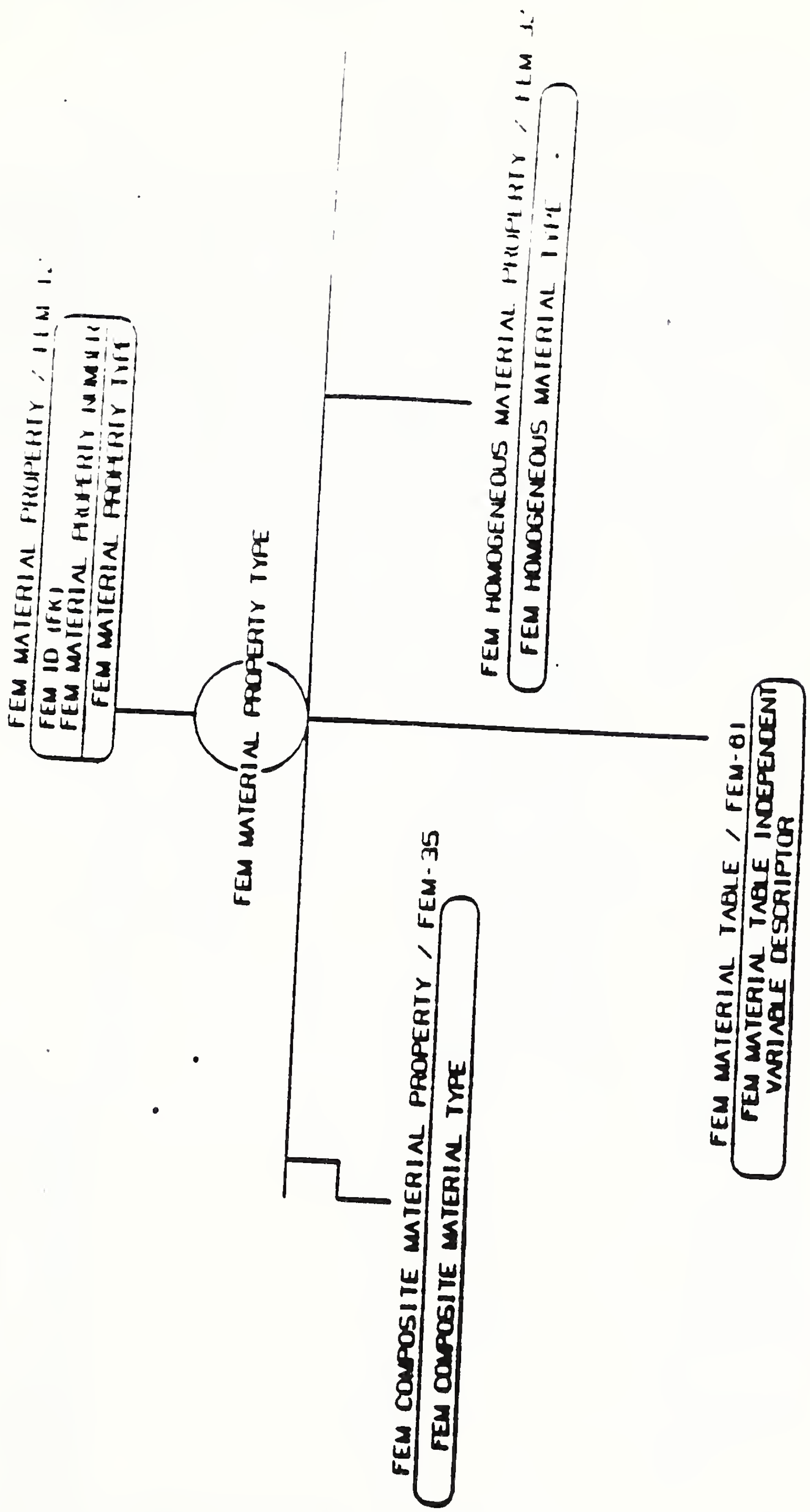

FEM MATERIAL PROPERTY / FEM
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
FEM MATERIAL PROPERTY TYPE
FEM HOMOGENEOUS MATERIAL PROPERTY / FEM
FEM HOMOGENEOUS MATERIAL TYPE
FEM MATERIAL TABLE / FEM-81
FEM MATERIAL TABLE INDEPENDENT VARIABLE DESCRIPTOR
FEM COMPOSITE MATERIAL PROPERTY / FEM-35
FEM COMPOSITE MATERIAL TYPE

Figure D-116: FEM Material Properties

FEM MATERIAL TABLE

Figure 118 represents the concept that a material's properties sometime vary with respect to independent variables. Typically, in structural mechanics problems where there is a wide variation of temperature among a FEM's nodes, material property data is input as a tabular set of information with the independent variable being temperature (or time) and the dependent variables being the material's Young's Modulus, Poisson's Ratio, etc... These data are interpolated with respect to temperature to determine what material property value to use.

Because the FEM MATERIAL TABLE data entity (along with its companion entity, FEM MATERIAL TABLE INSTANCE) recursively points to a FEM MATERIAL PROPERTY, any number of independent variables may be defined.

Rule FEM/R-15 apply to this IDEF1X diagram.

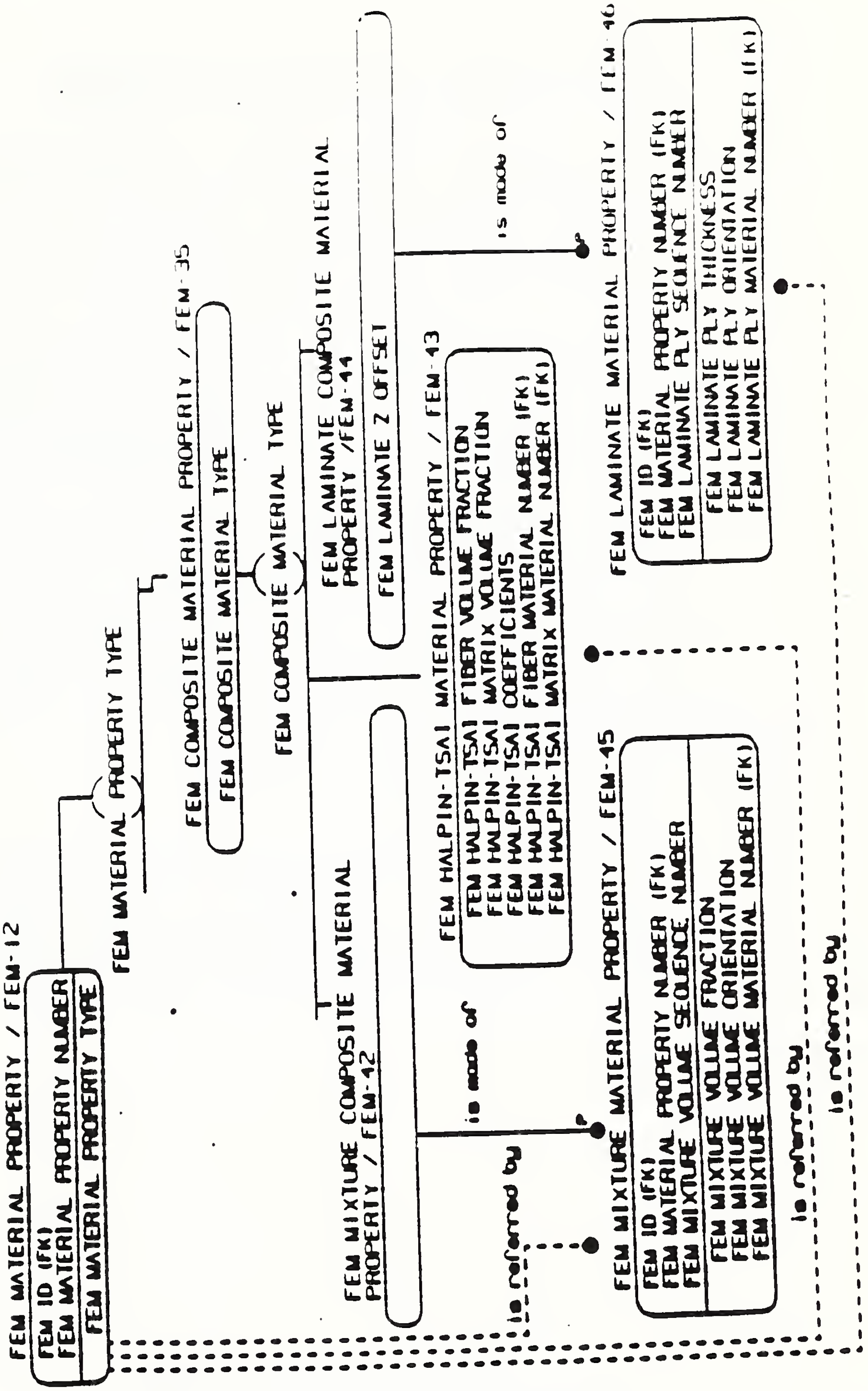

FEM MATERIAL PROPERTY / FEM-12
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
FEM MATERIAL PROPERTY TYPE
FEM COMPOSITE MATERIAL PROPERTY / FEM-35
FEM COMPOSITE MATERIAL TYPE
FEM COMPOSITE MATERIAL TYPE
FEM LAMINATE COMPOSITE MATERIAL PROPERTY /FEM-44
FEM LAMINATE Z OFFSET
FEM MIXTURE COMPOSITE MATERIAL PROPERTY / FEM-42
FEM HALPIN-TSAI MATERIAL PROPERTY / FEM-43
FEM HALPIN-TSAI FIBER VOLUME FRACTION
FEM HALPIN-TSAI MATRIX VOLUME FRACTION
FEM HALPIN-TSAI COEFFICIENTS
FEM HALPIN-TSAI FIBER MATERIAL NUMBER (FK)
FEM HALPIN-TSAI MATRIX MATERIAL NUMBER (FK)
is made of
FEM LAMINATE MATERIAL PROPERTY / FEM-46
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM LAMINATE PLY SEQUENCE NUMBER
FEM LAMINATE PLY THICKNESS
FEM LAMINATE PLY ORIENTATION
FEM LAMINATE PLY MATERIAL NUMBER (FK)
is made of
FEM MIXTURE MATERIAL PROPERTY / FEM-45
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM MIXTURE VOLUME SEQUENCE NUMBER
FEM MIXTURE VOLUME FRACTION
FEM MIXTURE VOLUME ORIENTATION
FEM MIXTURE VOLUME MATERIAL NUMBER (FK)
is referred by
is referred by
is referred by

Figure D-117: FEM Composite Material Property

FEM HOMOGENEOUS MATERIAL PROPERTY

The FEM HOMOGENEOUS MATERIAL PROPERTY shown in figure 119. That figure (119) displays three types of homogeneous material and they are:

1. the FEM ISOTROPIC MATERIAL PROPERTY, figure 120;
2. the FEM ORTHOTROPIC MATERIAL PROPERTY, figure 121;
3. the FEM ANISOTROPIC MATERIAL PROPERTY, figure 125;

Both the FEM ORTHOTROPIC MATERIAL PROPERTY, figure 121 and the FEM ANISOTROPIC MATERIAL PROPERTY, figure 125 are broken into three part. Those part are structural material properties, thermal material properties and thermal expansion material properties. The following is the list of those figures.

1. FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY, figure 122;
2. FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY, figure 123;
3. FEM ORTHOTROPIC THERMAL EXPANSION MATERIAL PROPERTY, figure 124;
4. FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY, figure 126;
5. FEM ANISOTROPIC THERMAL MATERIAL PROPERTY, figure 127;
6. FEM ANISOTROPIC THERMAL EXPANSION MATERIAL PROPERTY, figure 128;

Rule FEM/R-13 apply to 125 IDEF1X diagram.

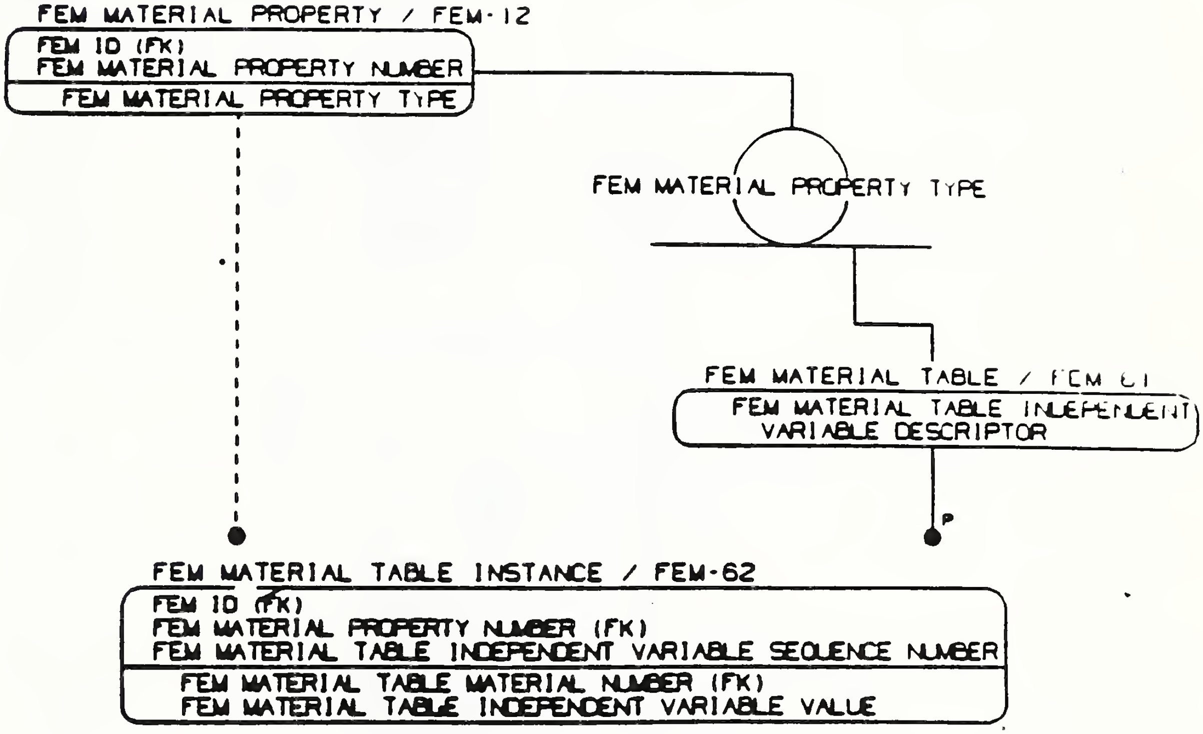

FEM MATERIAL PROPERTY / FEM-12
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM MATERIAL PROPERTY TYPE
FEM MATERIAL PROPERTY TYPE
FEM MATERIAL TABLE / FEM-61
FEM MATERIAL TABLE INDEPENDENT VARIABLE DESCRIPTOR
P
FEM MATERIAL TABLE INSTANCE / FEM-62
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM MATERIAL TABLE INDEPENDENT VARIABLE SEQUENCE NUMBER
FEM MATERIAL TABLE MATERIAL NUMBER (FK)
FEM MATERIAL TABLE INDEPENDENT VARIABLE VALUE

Figure D-118: FEM Material Table

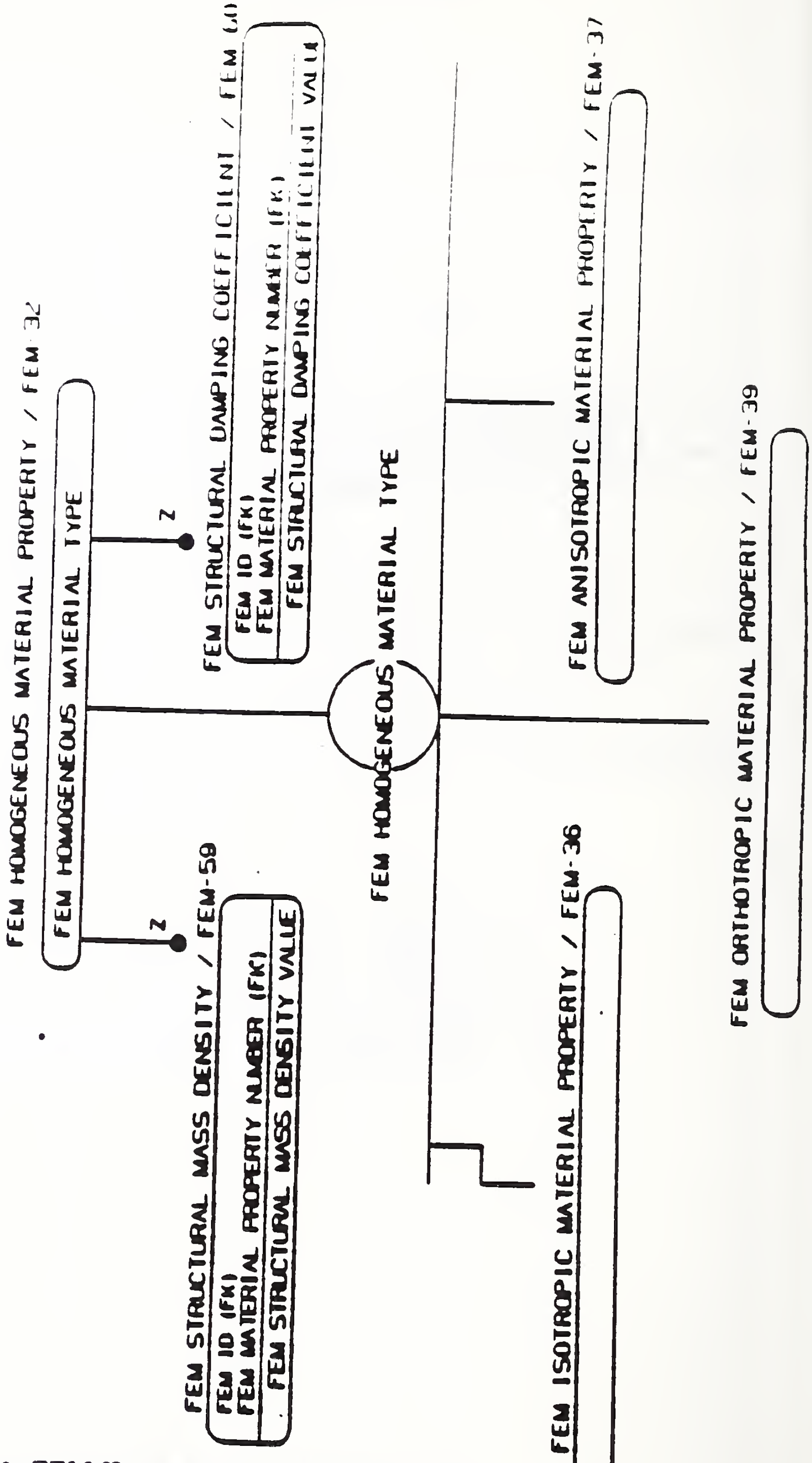

FEM HOMOGENEOUS MATERIAL PROPERTY / FEM-32
FEM HOMOGENEOUS MATERIAL TYPE
Z
FEM STRUCTURAL DAMPING COEFFICIENT / FEM-60
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM STRUCTURAL DAMPING COEFFICIENT VALUE
Z
FEM STRUCTURAL MASS DENSITY / FEM-59
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM STRUCTURAL MASS DENSITY VALUE
FEM HOMOGENEOUS MATERIAL TYPE
FEM ANISOTROPIC MATERIAL PROPERTY / FEM-37
FEM ISOTROPIC MATERIAL PROPERTY / FEM-36
FEM ORTHOTROPIC MATERIAL PROPERTY / FEM-39

Figure D-119: FEM Homogeneous Material Properties

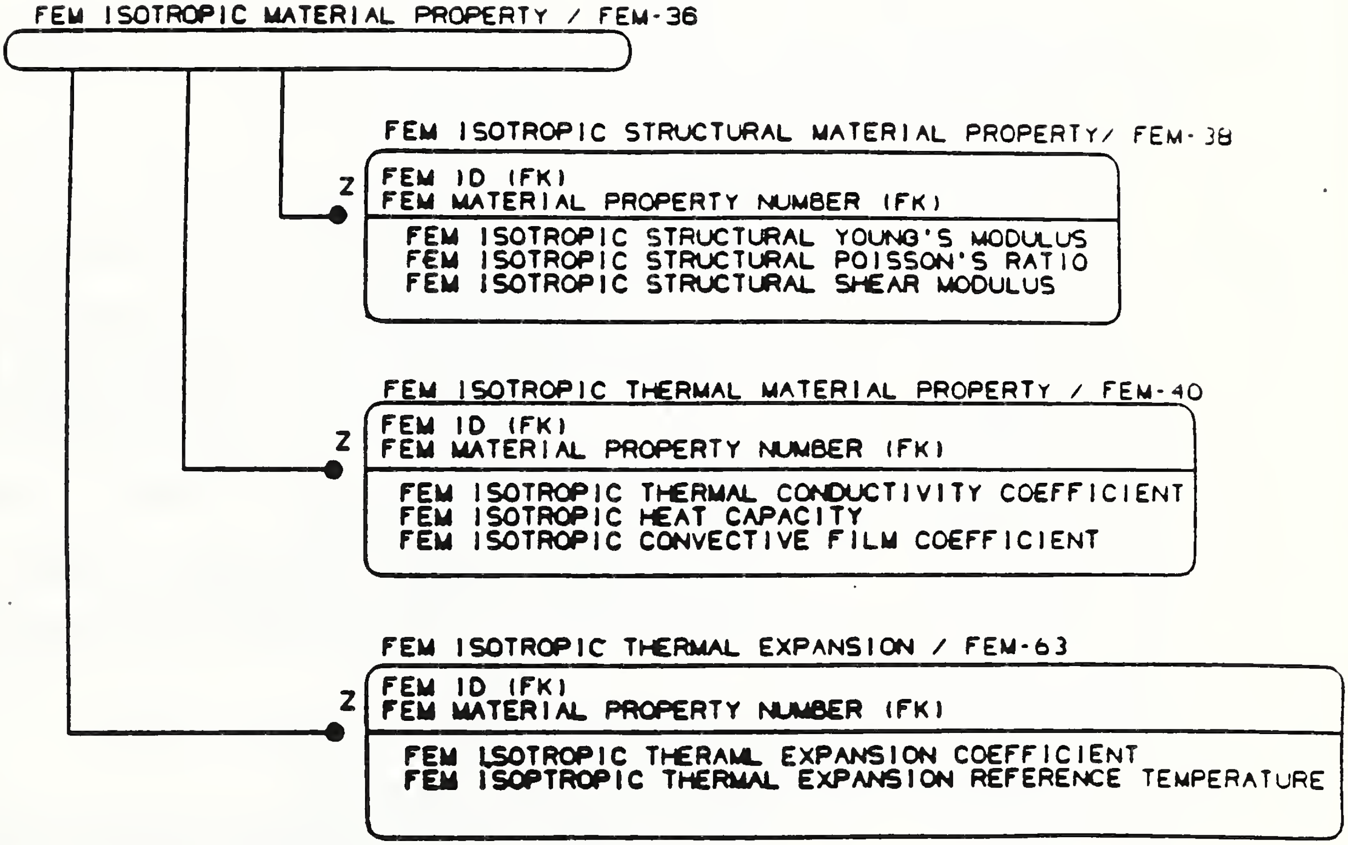

FEM ISOTROPIC MATERIAL PROPERTY / FEM-36
FEM ISOTROPIC STRUCTURAL MATERIAL PROPERTY/ FEM-38
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ISOTROPIC STRUCTURAL YOUNG'S MODULUS
FEM ISOTROPIC STRUCTURAL POISSON'S RATIO
FEM ISOTROPIC STRUCTURAL SHEAR MODULUS
FEM ISOTROPIC THERMAL MATERIAL PROPERTY / FEM-40
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ISOTROPIC THERMAL CONDUCTIVITY COEFFICIENT
FEM ISOTROPIC HEAT CAPACITY
FEM ISOTROPIC CONVECTIVE FILM COEFFICIENT
FEM ISOTROPIC THERMAL EXPANSION / FEM-63
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ISOTROPIC THERAML EXPANSION COEFFICIENT
FEM ISOPTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE

Figure D-120: FEM Isotropic Material Properties

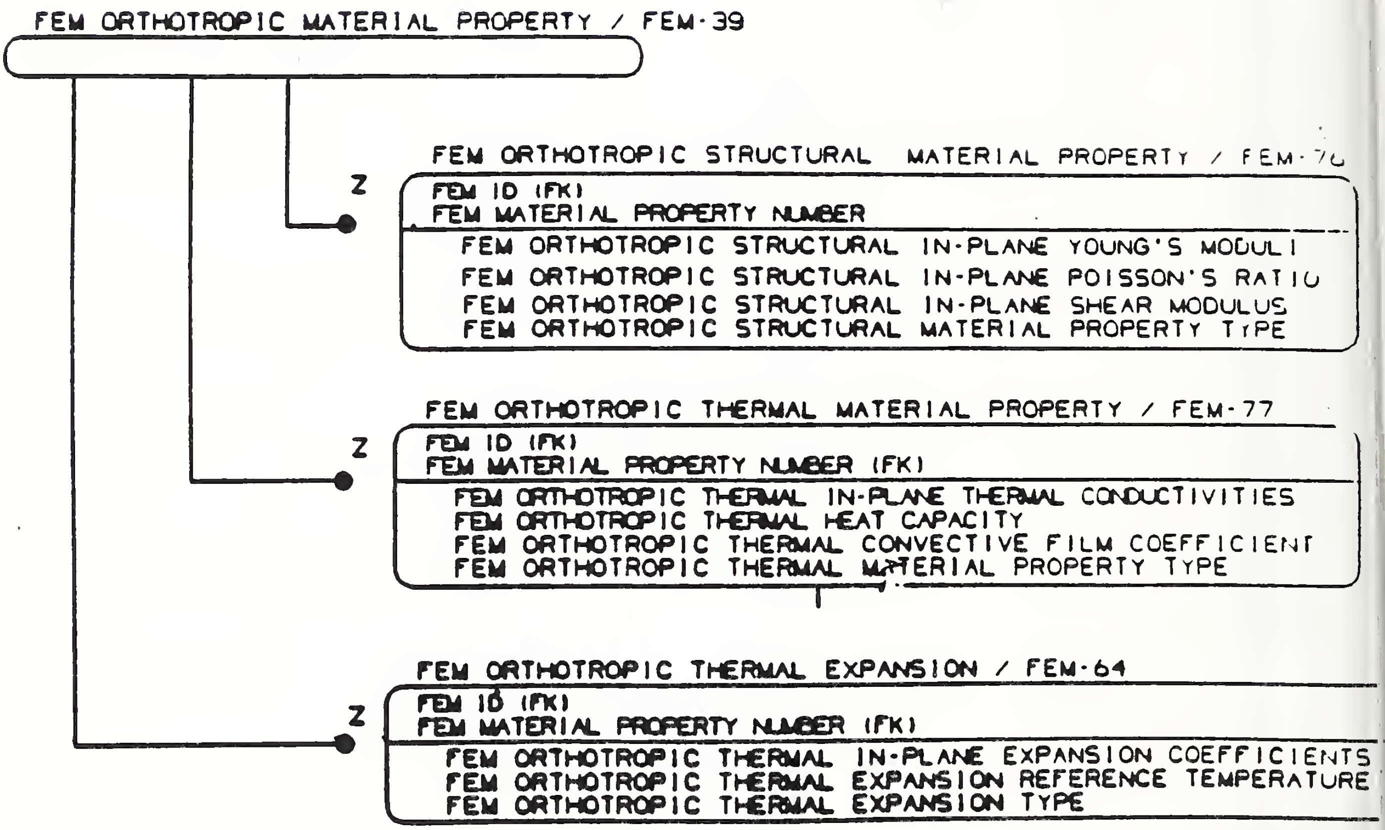

FEM ORTHOTROPIC MATERIAL PROPERTY / FEM-39
FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY / FEM-76
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ORTHOTROPIC STRUCTURAL IN-PLANE YOUNG'S MODULI
FEM ORTHOTROPIC STRUCTURAL IN-PLANE POISSON'S RATIO
FEM ORTHOTROPIC STRUCTURAL IN-PLANE SHEAR MODULUS
FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY / FEM-77
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ORTHOTROPIC THERMAL IN-PLANE THERMAL CONDUCTIVITIES
FEM ORTHOTROPIC THERMAL HEAT CAPACITY
FEM ORTHOTROPIC THERMAL CONVECTIVE FILM COEFFICIENT
FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC THERMAL EXPANSION / FEM-64
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ORTHOTROPIC THERMAL IN-PLANE EXPANSION COEFFICIENTS
FEM ORTHOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE
FEM ORTHOTROPIC THERMAL EXPANSION TYPE

Figure D-121: FEM Orthotropic Material Properties

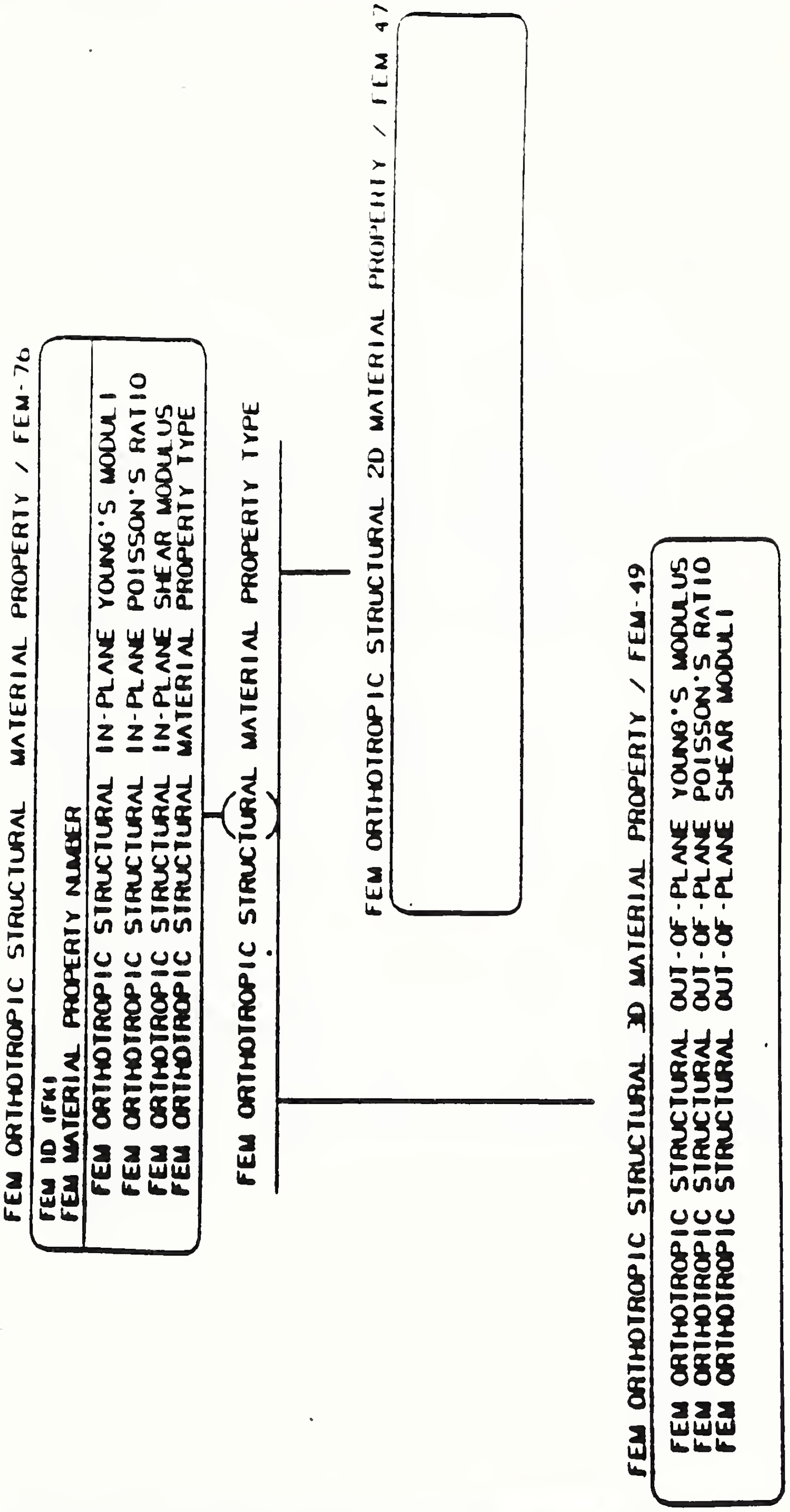

FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY / FEM-76
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ORTHOTROPIC STRUCTURAL IN-PLANE YOUNG'S MODULI
FEM ORTHOTROPIC STRUCTURAL IN-PLANE POISSON'S RATIO
FEM ORTHOTROPIC STRUCTURAL IN-PLANE SHEAR MODULUS
FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC STRUCTURAL 2D MATERIAL PROPERTY / FEM-47
FEM ORTHOTROPIC STRUCTURAL 3D MATERIAL PROPERTY / FEM-49
FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE YOUNG'S MODULUS
FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE POISSON'S RATIO
FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE SHEAR MODULI

Figure D-122: FEM Orthotropic Structural Material Property

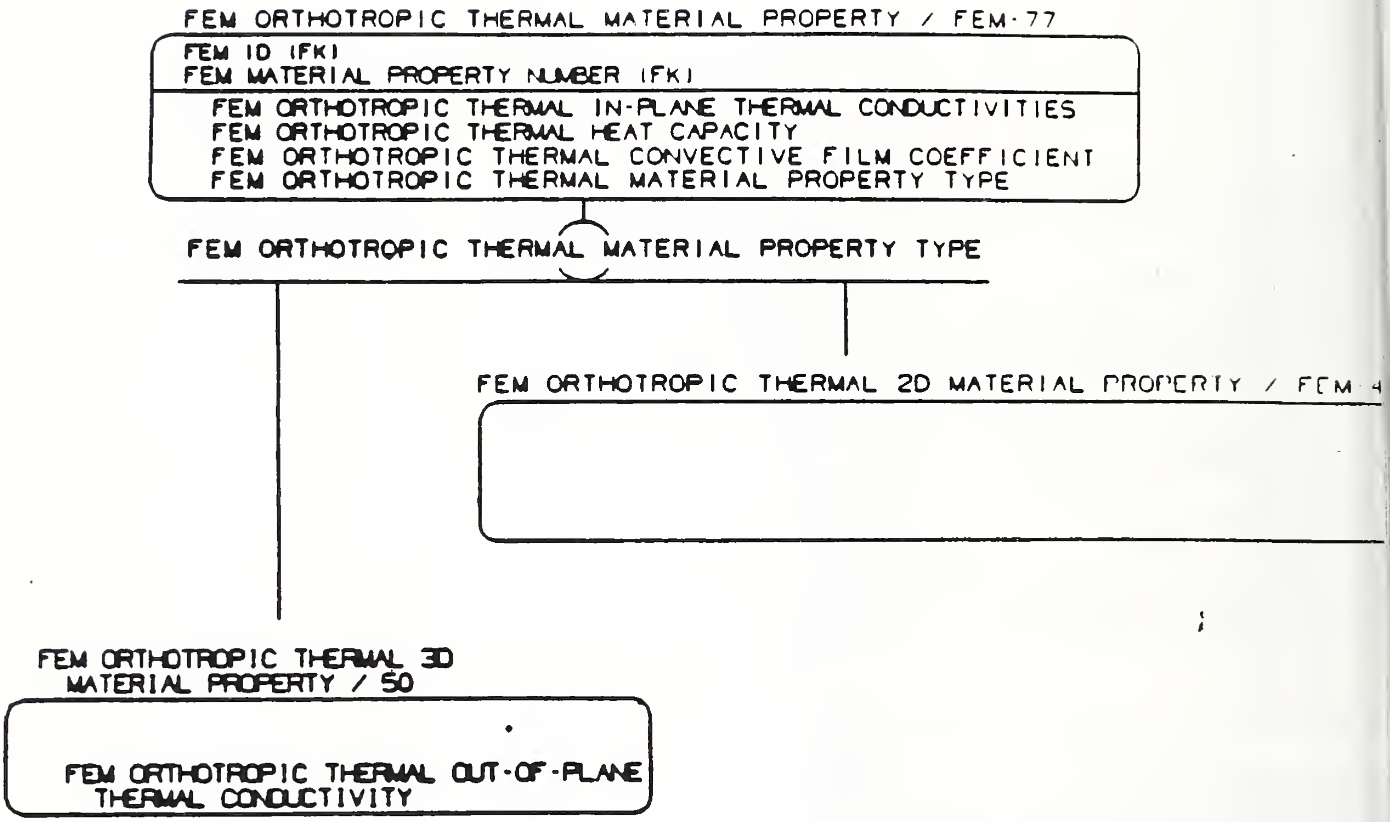

FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY / FEM-77
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ORTHOTROPIC THERMAL IN-PLANE THERMAL CONDUCTIVITIES
FEM ORTHOTROPIC THERMAL HEAT CAPACITY
FEM ORTHOTROPIC THERMAL CONVECTIVE FILM COEFFICIENT
FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY TYPE
FEM ORTHOTROPIC THERMAL 2D MATERIAL PROPERTY / FEM-4
FEM ORTHOTROPIC THERMAL 3D MATERIAL PROPERTY / 50
FEM ORTHOTROPIC THERMAL OUT-OF-PLANE THERMAL CONDUCTIVITY

Figure D-123: FEM Orthotropic Thermal Material Property

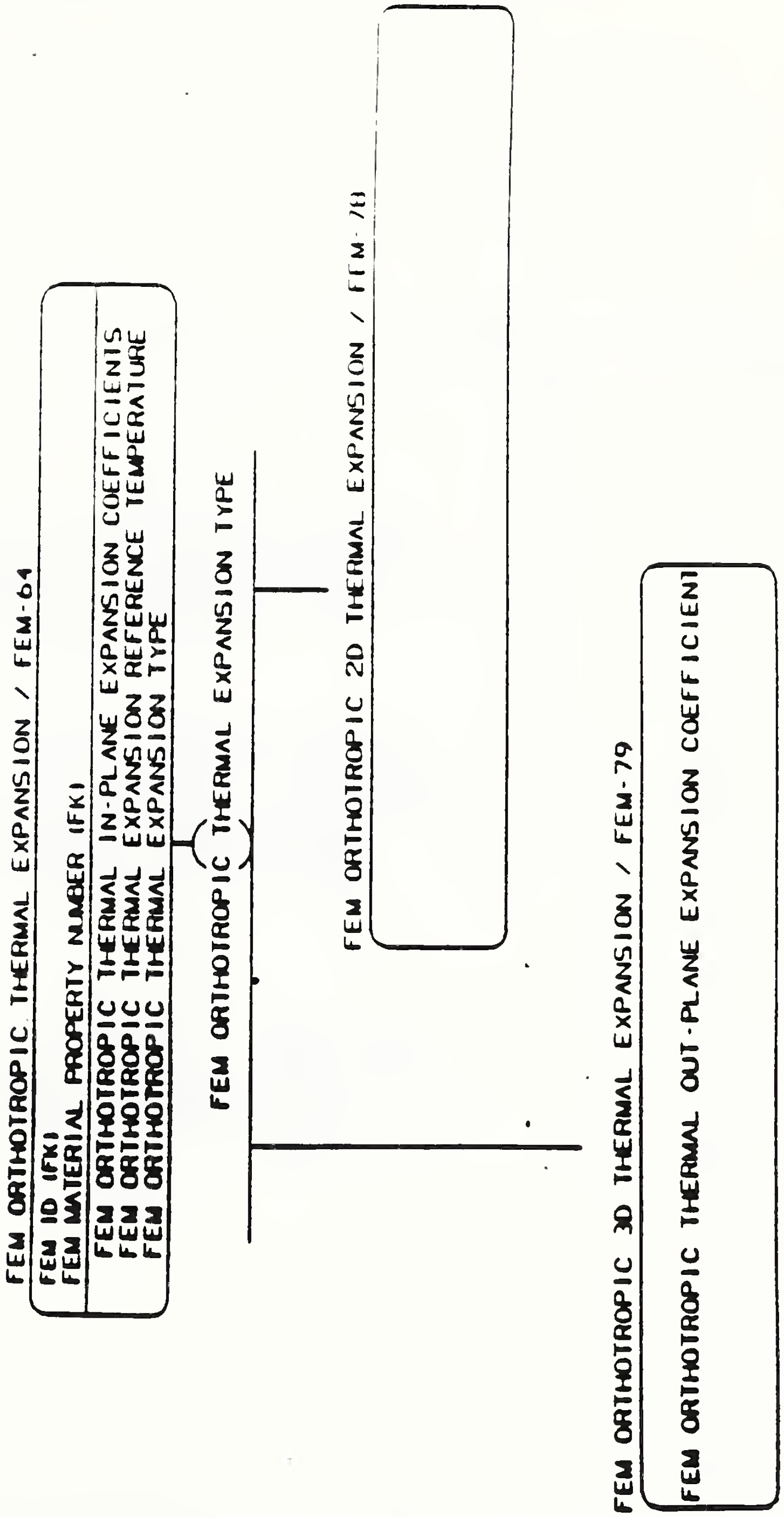

FEM ORTHOTROPIC THERMAL EXPANSION / FEM-64
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER (FK)
FEM ORTHOTROPIC THERMAL IN-PLANE EXPANSION COEFFICIENTS
FEM ORTHOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE
FEM ORTHOTROPIC THERMAL EXPANSION TYPE
FEM ORTHOTROPIC THERMAL EXPANSION TYPE
FEM ORTHOTROPIC 2D THERMAL EXPANSION / FEM-78
FEM ORTHOTROPIC 3D THERMAL EXPANSION / FEM-79
FEM ORTHOTROPIC THERMAL OUT-PLANE EXPANSION COEFFICIENT

Figure D-124: FEM Orthotropic Thermal Expansion Material Property

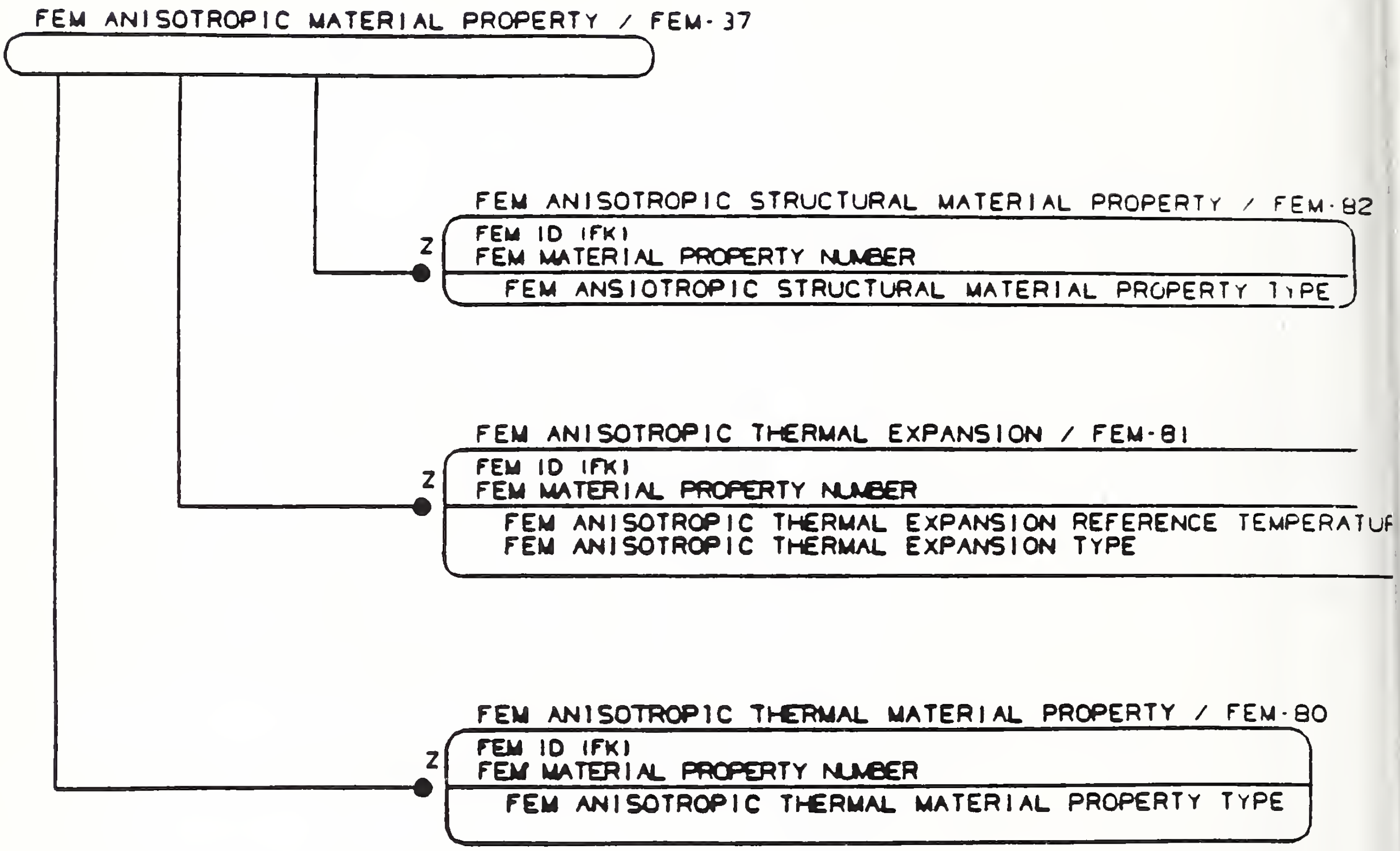

FEM ANISOTROPIC MATERIAL PROPERTY / FEM-37
FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY / FEM-82
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ANISOTROPIC THERMAL EXPANSION / FEM-81
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC THERMAL EXPANSION REFERENCE TEMPERATUR
FEM ANISOTROPIC THERMAL EXPANSION TYPE
FEM ANISOTROPIC THERMAL MATERIAL PROPERTY / FEM-80
Z
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC THERMAL MATERIAL PROPERTY TYPE

Figure D-125: FEM Anisotropic Material Property

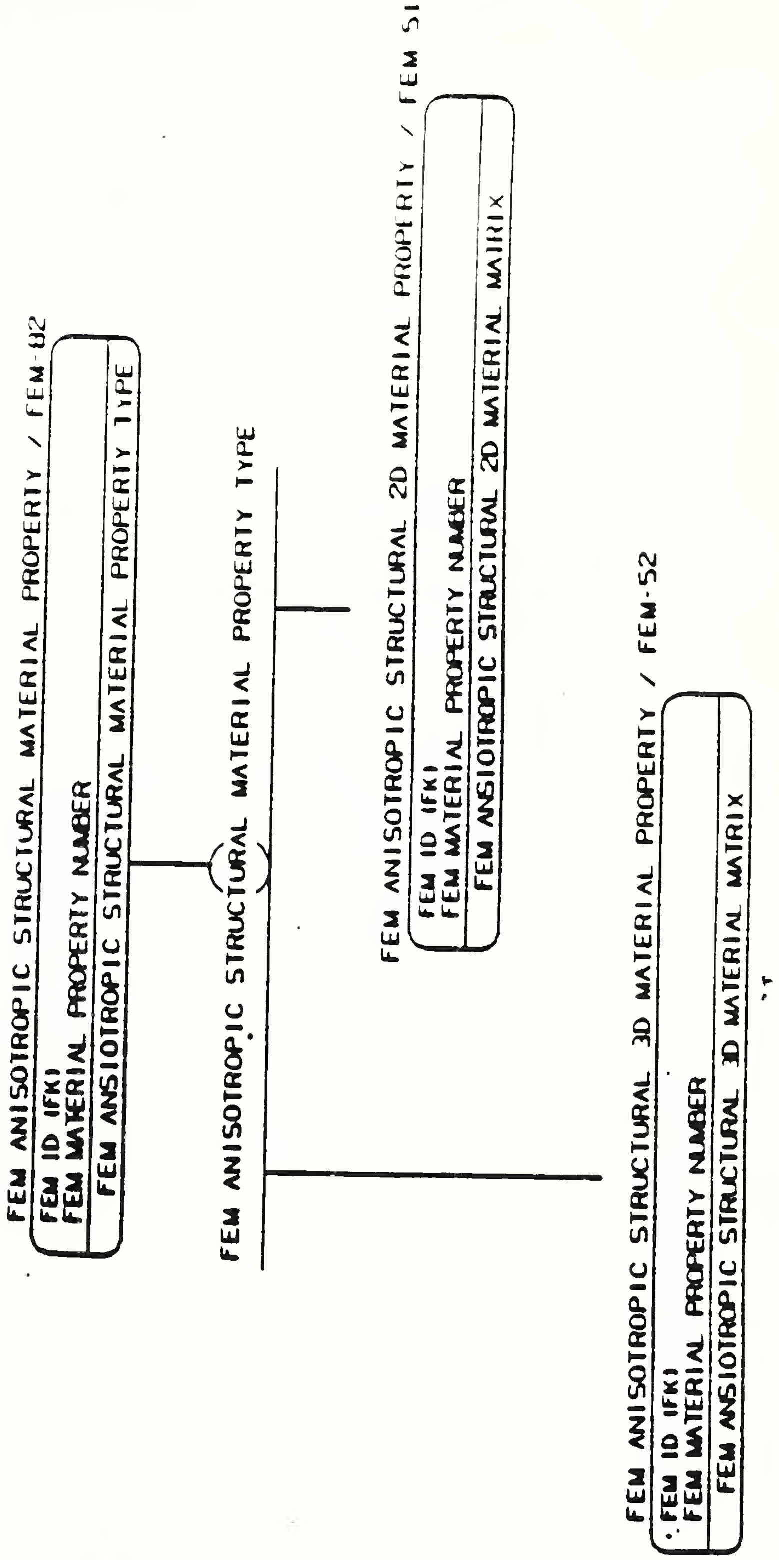

FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY / FEM-82
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE
FEM ANISOTROPIC STRUCTURAL 2D MATERIAL PROPERTY / FEM-51
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC STRUCTURAL 2D MATERIAL MATRIX
FEM ANISOTROPIC STRUCTURAL 3D MATERIAL PROPERTY / FEM-52
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC STRUCTURAL 3D MATERIAL MATRIX

Figure D-126: FEM Anisotropic Structural Material Property

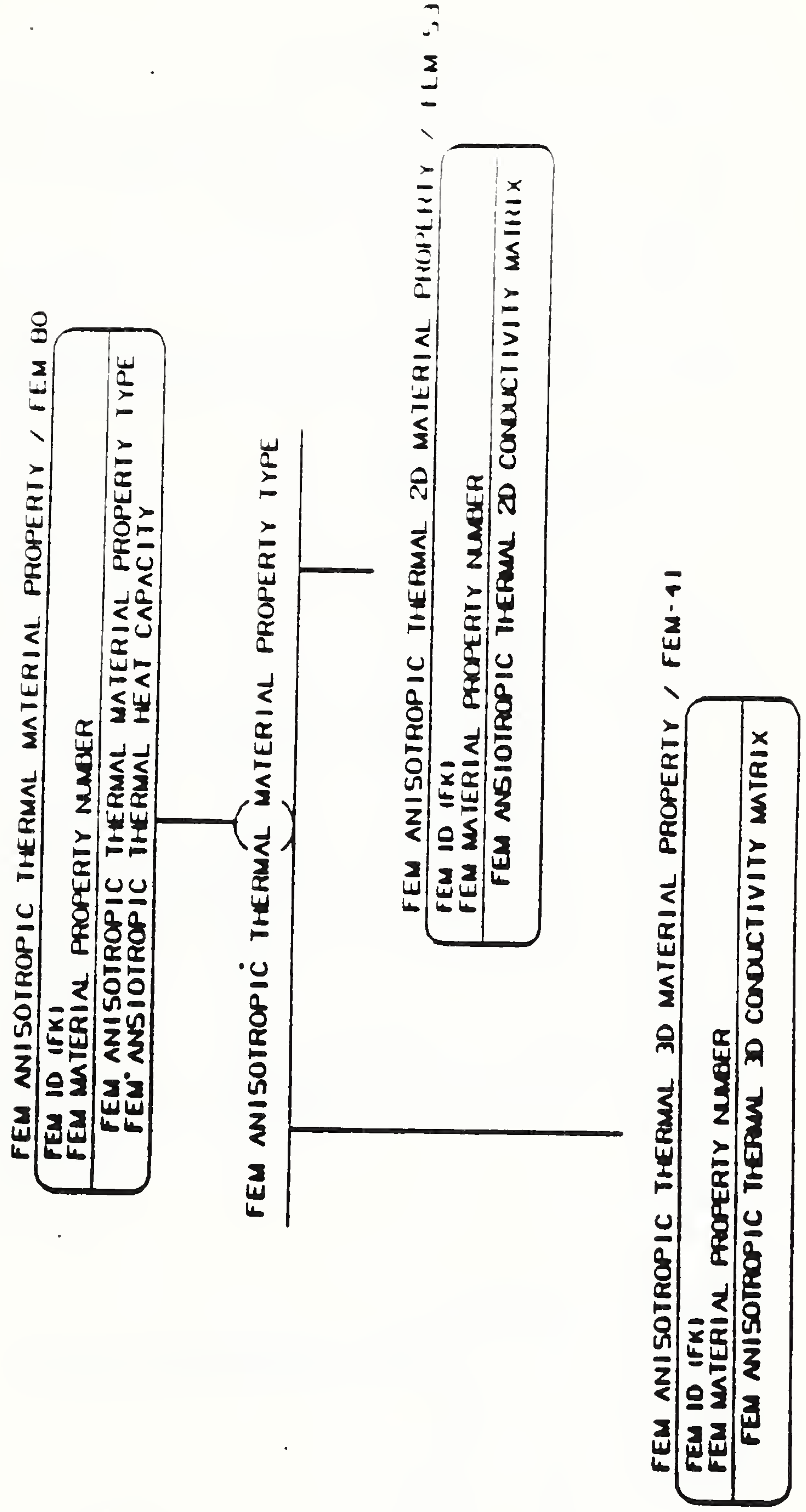

FEM ANISOTROPIC THERMAL MATERIAL PROPERTY / FEM 80
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC THERMAL MATERIAL PROPERTY TYPE
FEM ANSIOTROPIC THERMAL HEAT CAPACITY
FEM ANISOTROPIC THERMAL MATERIAL PROPERTY TYPE
FEM ANISOTROPIC THERMAL 2D MATERIAL PROPERTY / FEM 53
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC THERMAL 2D CONDUCTIVITY MATRIX
FEM ANISOTROPIC THERMAL 3D MATERIAL PROPERTY / FEM-41
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC THERMAL 3D CONDUCTIVITY MATRIX

Figure D-127: FEM Anisotropic Thermal Material Property

9.3.3 FEM Entity Definitions

The following sections describe each data entity in the FEM reference model. These sections are presented in numerical order by entity number.

Entity Name: **FEA CONTROL**

Entity Number: **FEAC-1**

Ancillary data pertaining to a finite element analysis. Load cases are collected into FEM ENVIRONMENTS and then applied to a FEM via a FEA CONTROL.

Primary Key Attributes:

FEA CONTROL ID — Data Type: Integer

This attribute contains the unique identifier of a FEA CONTROL.

Other Attributes:

FEA CONTROL ANALYSIS ASSUMPTION — Data Type: String

Text describing the analysis assumption used by the analyst in making the FEA CONTROL.

FEA CONTROL ANALYSIS TYPE — Data Type: String

Text describing the type of analysis that the FEA CONTROL was created for. For example, a value of this attribute might be 'plane-stress'.

FEA CONTROL TARGETED ANALYSIS CODE — Data Type: String

This attribute contains the target analysis code that the FEA CONTROL was created for.

FEA CONTROL UNITS ID (FK) — Data Type: String

The identifier of the fundamental units that will be used in the FEA CONTROL.
See attribute definition for UNIT ID in entity UNITS [Entity UNIT-1].

Business rules:

EXPRESS Specification:

There is currently no EXPRESS specification for this entity.

Entity Name: **FEA**

Entity Number: **FEA-1**

Application of a FEA CONTROL to a FEM, via a computer program, creates a finite element analysis (FEA).

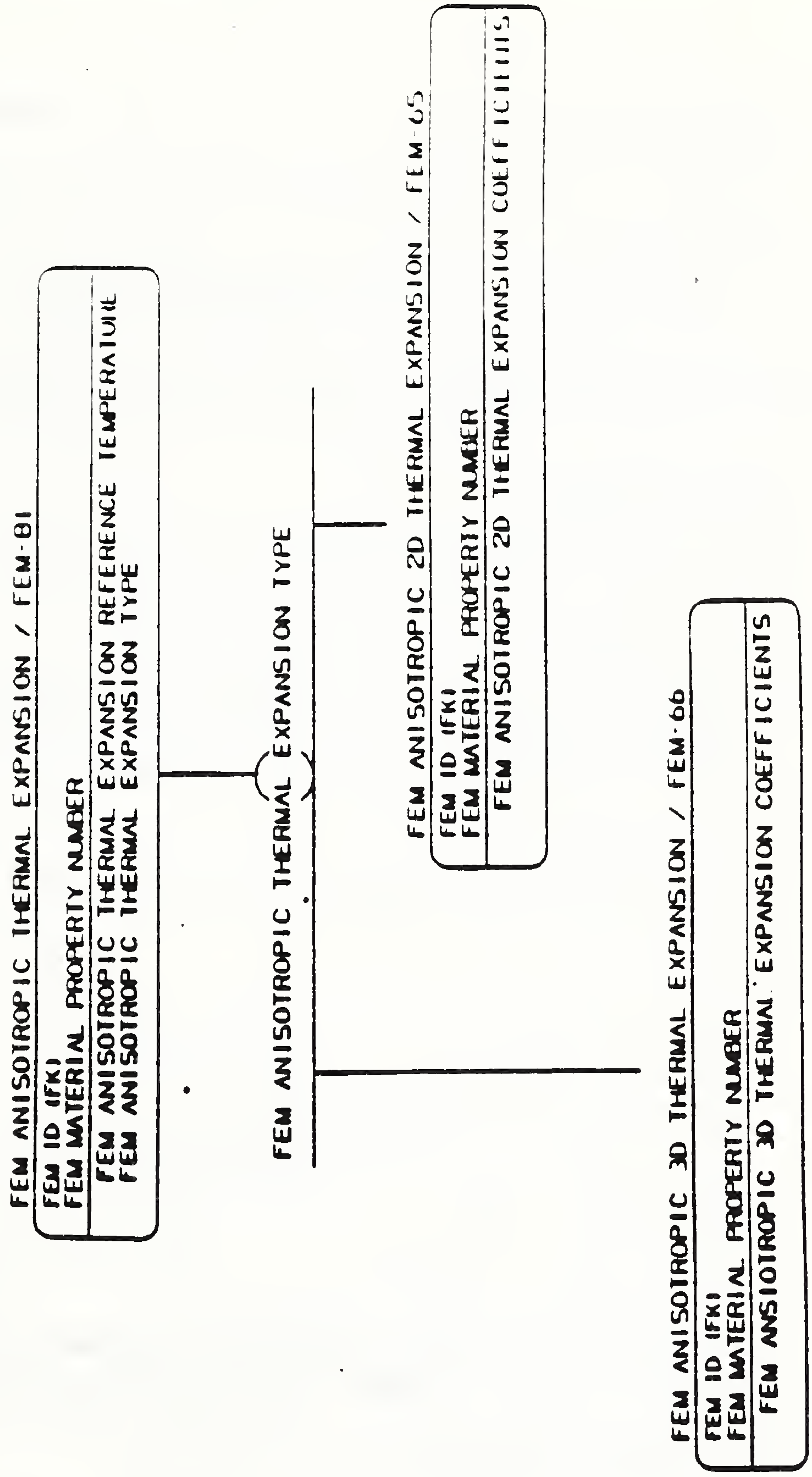

FEM ANISOTROPIC THERMAL EXPANSION / FEM-81
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE
FEM ANISOTROPIC THERMAL EXPANSION TYPE
FEM ANISOTROPIC THERMAL EXPANSION TYPE
FEM ANISOTROPIC 2D THERMAL EXPANSION / FEM-65
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANISOTROPIC 2D THERMAL EXPANSION COEFFICIENTS
FEM ANISOTROPIC 3D THERMAL EXPANSION / FEM-66
FEM ID (FK)
FEM MATERIAL PROPERTY NUMBER
FEM ANSIOTROPIC 3D THERMAL EXPANSION COEFFICIENTS

Figure D-128: FEM Anisotropic Thermal Expansion Material Property

Primary Key Attributes:

FEA ID	Data Type: Integer

The unique identifier of a FEA.

FEM ID (FK)	Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEA CONTROL ID (FK)	Data Type: Integer

See attribute defination in entity FEA CONTROL [Entity FEM-21].

Other Attributes:

FEA ACTUAL ANALYSIS CODE	Data Type: String

This attribute contains the of the actual analysis code that was used in the finite element analysis.

FEA AUTHOR	Data Type: String

This attribute contains the analysis creator's name or names.

FEA DESCRIPTOR	Data Type: String

Text that describes the FEA and provides supporting information supplied by the analyst.

FEA UNITS ID	Data Type: String

The identifier of the fundamental units that will be used in the FEA.

Business rules:

EXPRESS specification:

There is currently no EXPRESS for this entity.

Entity Name: **FEM**

Entity Number: **FEM-1**

A FEM is a collection of finite elements, nodes, material properties, and geometric characteristics which is used to mathematically model the physical behavior of a thing.

Primary Key Attributes:

FEM ID	Data Type: String

This attribute represents the unique identifier of a FEM.

Other Attributes:

FEM AUTHOR Data Type: String

This attribute contains the FEM creator's name or names.

FEM CREATING SOFTWARE Data Type: String

This attribute contains the preprocessor used to create the FEM. This is normally the name of the software used to create the FEM.

FEM CREATION DATE Data Type: Date

The date the FEM was created. For example, a possible value might be '31 March 1967'.

FEM CREATION TIME Data Type: Time

The time the FEM was created. For example, a possible value might be '1PM'.

FEM DESCRIPTOR Data Type: String

Text that describes the FEM and provides supporting information supplied by the analyst.

FEM TARGETED ANALYSIS CODE Data Type: String

This attribute contains the target analysis code that the FEM was created for.

FEM UNITS ID (FK) Data Type: String

The identifier of the fundamental units that will be used in the FEM.
See attribute defination in entity UNITS [Entity UNIT-1].

Business rules:

EXPRESS specification:

```
ENTITY finite_element_model;
  fem_id          : UNIQUE string;
  author          : INTERNAL person_name;
  creating_software      : string;
  creation_date   : INTERNAL date;
  creation_time   : INTERNAL time;
  descriptor      : string;
  targeted_analysis_code         : string;
  units_ref       : EXTERNAL units;
  product_item_ref       : OPTIONAL EXTERNAL SET [1:#] OF
                  product_item;
  master_coordinate_system_ref   : EXTERNAL master_coordinate_system;
  derived_coordinate_system_ref  : OPTIONAL EXTERNAL SET [1:#] OF
                  derived_coordinate_system;
  element_ref     : EXTERNAL SET [1:#] OF element;
  node_ref        : EXTERNAL SET [1:#] OF node;
  geometric_property_ref         : EXTERNAL SET [1:#] OF
```

```
                      geometric_property;
  material_property_ref : EXTERNAL SET [1:#] OF
                      material_property;
  approval_ref  : OPTIONAL EXTERNAL SET  [1:#] OF
                      approval;
  group : OPTIONAL EXTERNAL SET [1:#] OF
                      group;
END_ENTITY;

RULE fem_identifiers FOR (finite_element_model);
  LOCAL
    coord_sys : LIST [1 : #] OF fem_coordinate_system;
  END_LOCAL;
  coord_sys = coord_sys + master_coordinate_system_ref;
  REPEAT FOR EACH derived_coordinate_system IN
       derived_coordinate_system_ref;
    coord_sys = coord_sys + derived_coordinate_system;
  END_REPEAT;
  REPEAT i := 1 TO SIZEOF(coord_sys);
    IF NOT (UNIQUE coord_sys.coordinate_system_number) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH element IN element_ref;
    IF NOT (UNIQUE element.element_number) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH node IN node_ref;
    IF NOT (UNIQUE node.node_number) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH geometric_property IN geometric_property_ref;
    IF NOT (UNIQUE geometric_property.geometric_property_id) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH material_property IN material_property_ref;
    IF NOT (UNIQUE material_property.material_property_id) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH group IN group_ref;
    IF NOT (UNIQUE group.group_name) THEN
        VIOLATION;
    END_IF;
  END_REPEAT;
END_RULE;
```

Entity Name: **PRODUCT ITEM/FEM**

Entity Number: **FEM-2**

A PRODUCT ITEM VERSION is an entity that the enterprise keeps track of; e.g., a part, a drawing, an assembly, etc...The PRODUCT ITEM/FEM intersection data entity records which PRODUCT ITEM VERSIONs or pieces of PRODUCT ITEM VERSION the Finite Element Model FEM references and what FEMs have been used to describe a PRODUCT ITEM.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [entity FEM-1].

PRODUCT ITEM ID (FK) Data Type: String
This attribute contains the unique identifier for a PRODUCT ITEM.
See attribute defination in entity PRODCUT ITEM VERSION [Entity PSCM-4] for complete details.

Other Attributes:

Business rules:

EXPRESS specification:

The EXPRESS specification is missing.

Entity Name: **FEM CONNECTIVITY**

Entity Number: **FEM-6**

The FEM CONNECTIVITY data entity represents an order list of FEM NODEs that were used in the defination of a FEM ELEMENT. Some of the important features are:

1. The order list does not have to be complete, i.e., there can be gaps or missing FEM NODEs.
2. The position of a FEM NODE in the order list is very important in that it define the gemetric shape of the element. (See the figures 129 through 138 related to this entity.)

3. An order list of FEM NODE contected with an FEM ELEMENT is the normal manner that most FEM cumputer codes handle FEM CONNECTIVITY. Because of the requirements of the IDEF1X modeling methodology, the FEM CONNECTIVITY data entity was created in order to resolves the many-to-many relationship that exists between the FEM NODE and FEM ELEMENT data entities. The attribute FEM NODE SEQUENCE NUMBER was invented to satisfy the requirements of uniquiely identfying an occurence of this entity (or the postion in a FEM ELEMENTs order list). It is not required to be transferred in the physical file for an ordered list of a finite element's nodes will suffice in the EXPRESS entity for FEM ELEMENTs.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

FEM ELEMENT NUMBER (FK) Data Type: Integer

A number used to identify a finite element from all others within a FEM. It is only unique within the current FEM.

See attribute defination in FEM ELEMENT [Entity FEM-17].

FEM NODE SEQUENCE NUMBER Data Type: Integer

The FEM NODE SEQUENCE NUMBER indentfies the position of a FEM NODE in the element's sequence of FEM NODEs (an order list). This position is very important in that it define the gemetric shape of the element.

The following are figure defining the relationship of FEM NODE SEQUENCE NUMBER to FEM element of varies types (element shape number and element order number):

Figure Number	Reference Model Name
129	FEM Bar Node Connectivity
130	FEM Triangle Node Connectivity
131	FEM Quadrilateral Node Connectivity
132	FEM Tetrahedron Node Connectivity
133	FEM Wedge Node Connectivity, Order 0 and 1
134	FEM Wedge Node Connectivity, Order 2
135	FEM Prism Node Connectivity, Order 0 and 1
136	FEM Prism Node Connectivity, Order 2
137	FEM Hexahedron Node Connectivity, Order 0 and 1
138	FEM Hexahedron Node Connectivity, Order 2

Other Attributes:

FEM NODE NUMBER (FK) Data Type: Integer

A number used to identify a FEM node from all others within a FEM. It is only unique within the current FEM.

See attribute defination in entity FEM NODE [Entity FEM-16].

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1x in this order list of FEM NODES represented by this IDEF1X entity was move into the FEM ELEMENT EXPRESS entity. Therefore see entity FEM ELEMENT [Entity FEM-17] for details.

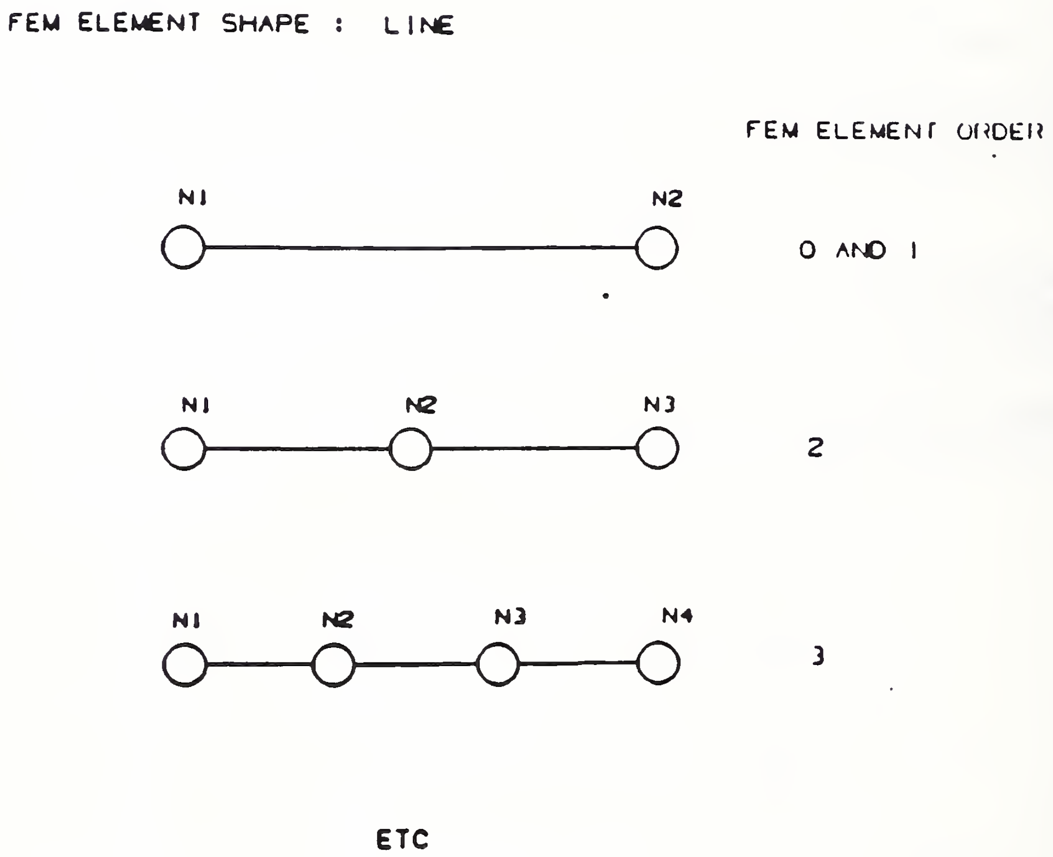


Figure D-129: FEM Line Node Connectivity

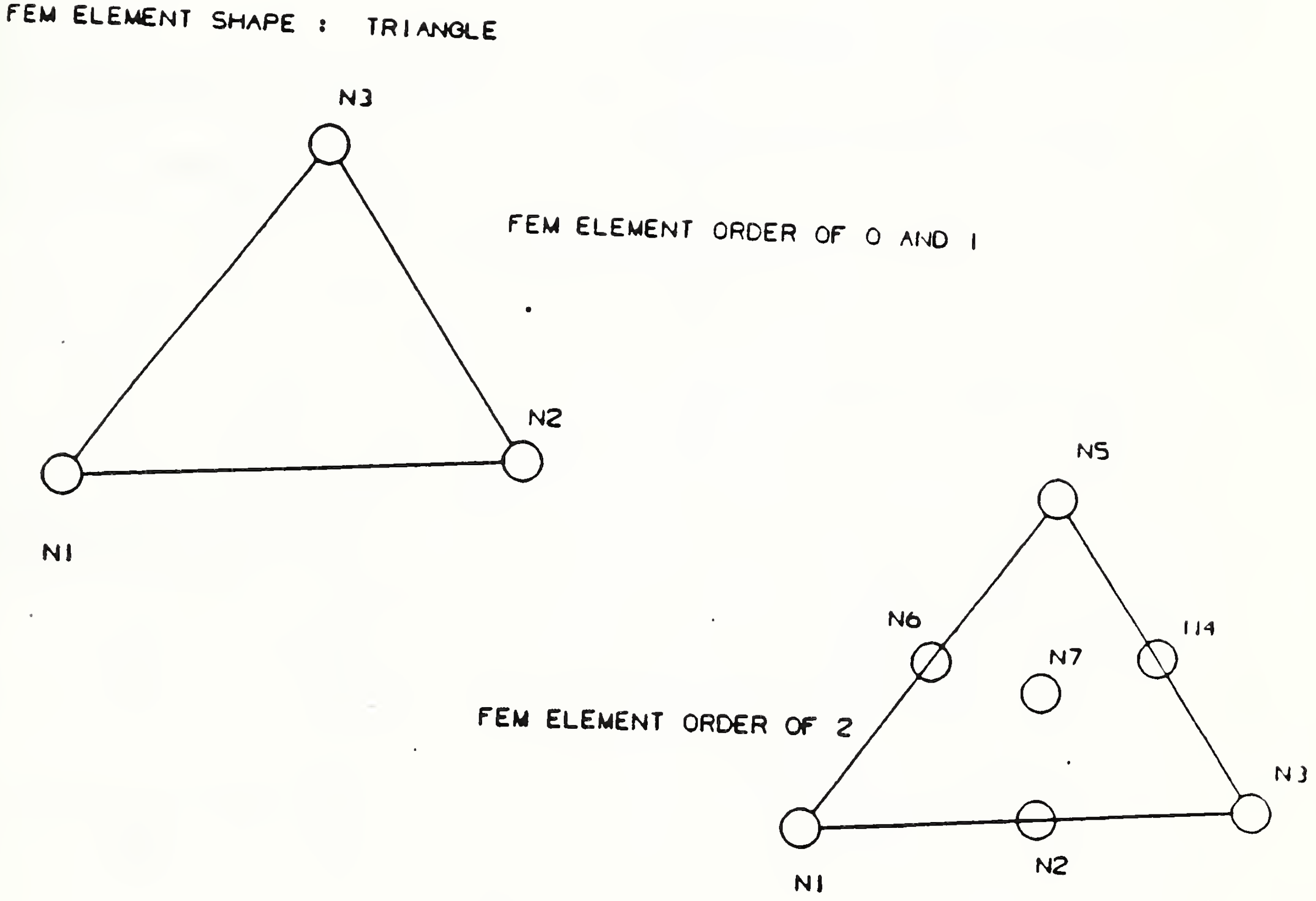


Figure D-130: FEM Triangle Node Connectivity

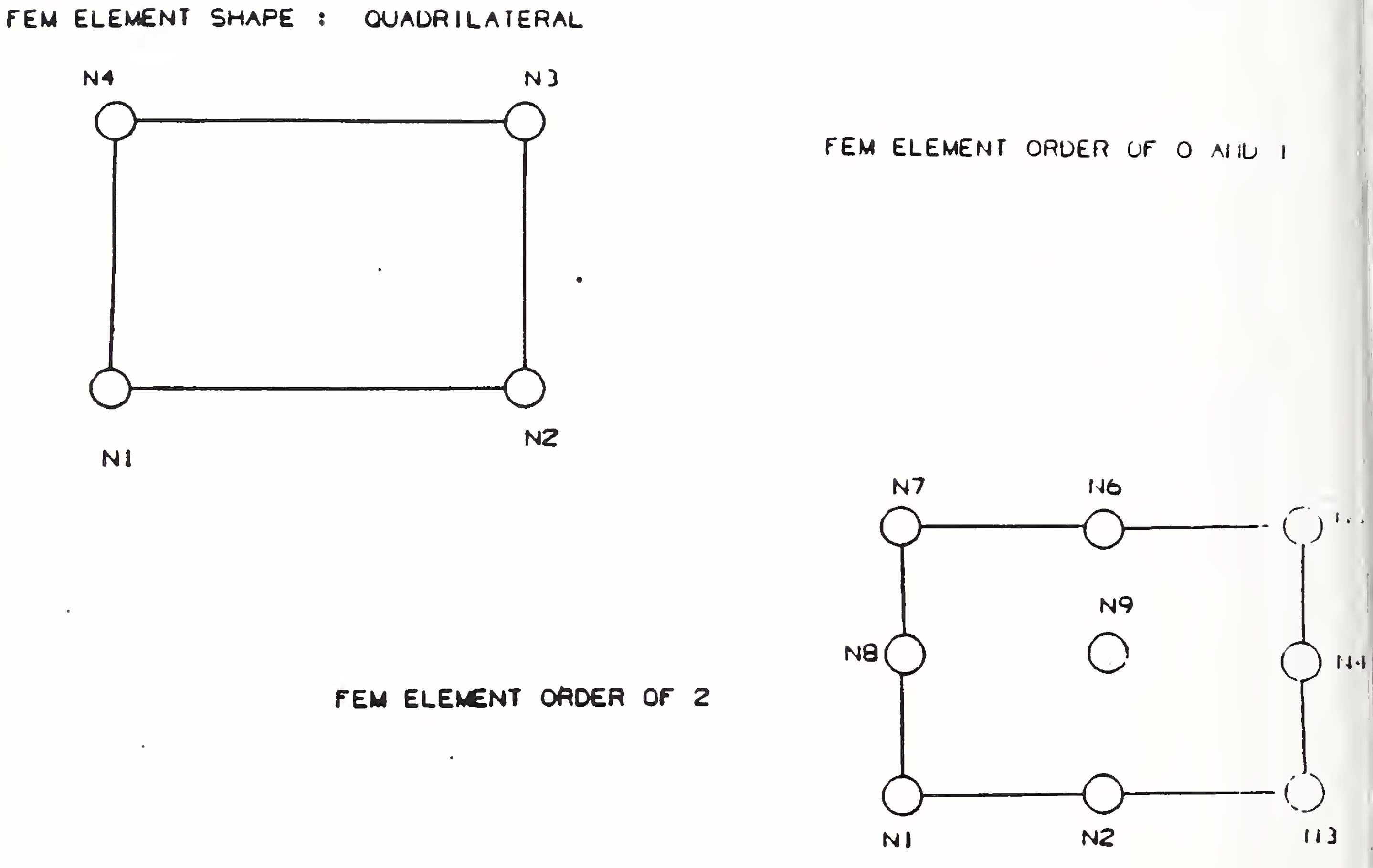

FEM ELEMENT SHAPE : QUADRILATERAL
N4
N3
N1
N2
FEM ELEMENT ORDER OF 0 AND 1
FEM ELEMENT ORDER OF 2
N7
N6
N9
N8
N4
N1
N2
N3

Figure D-131: FEM Quadrilateral Node Connectivity

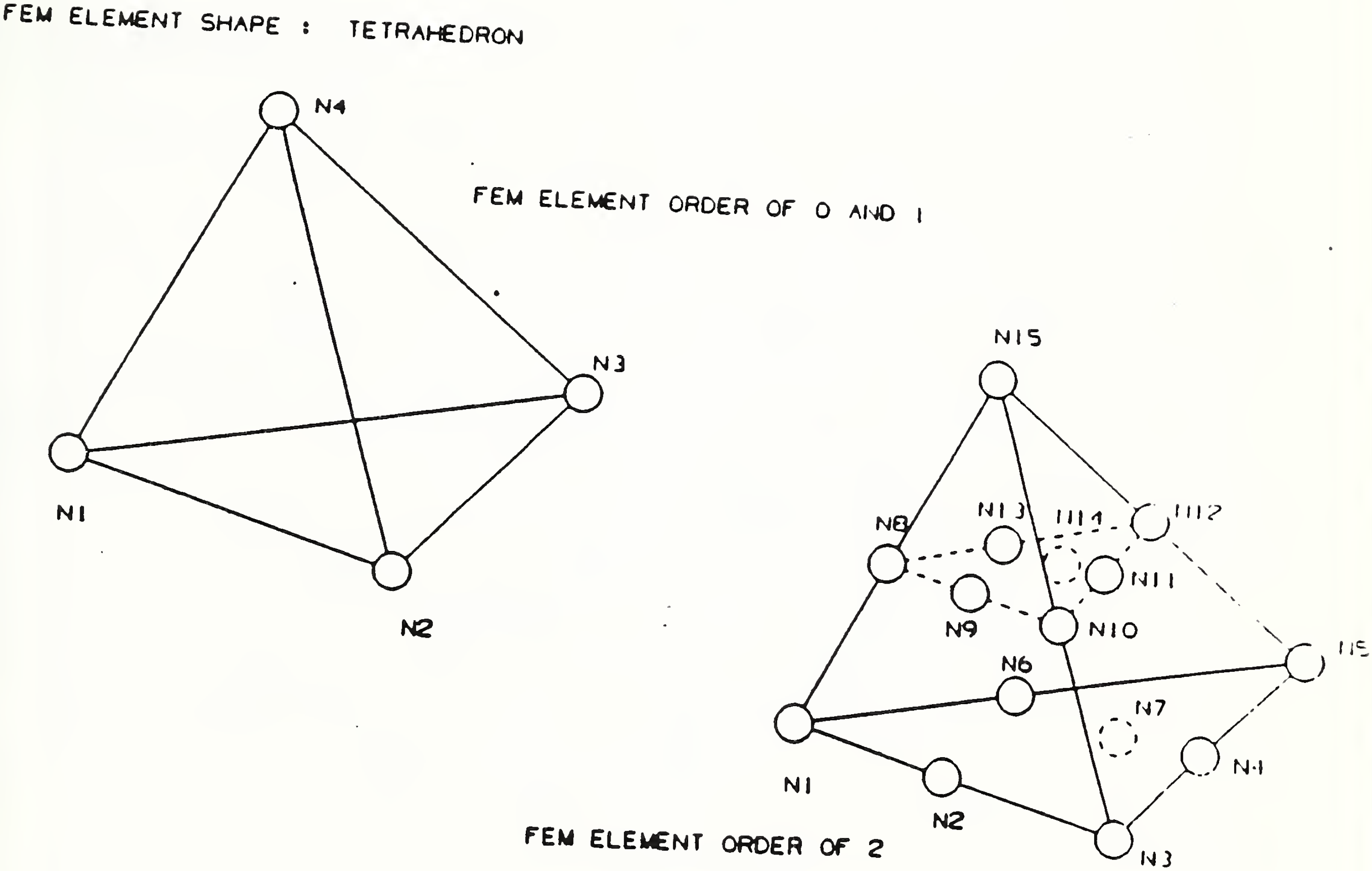


Figure D-132: FEM Tetrahedron Node Connectivity

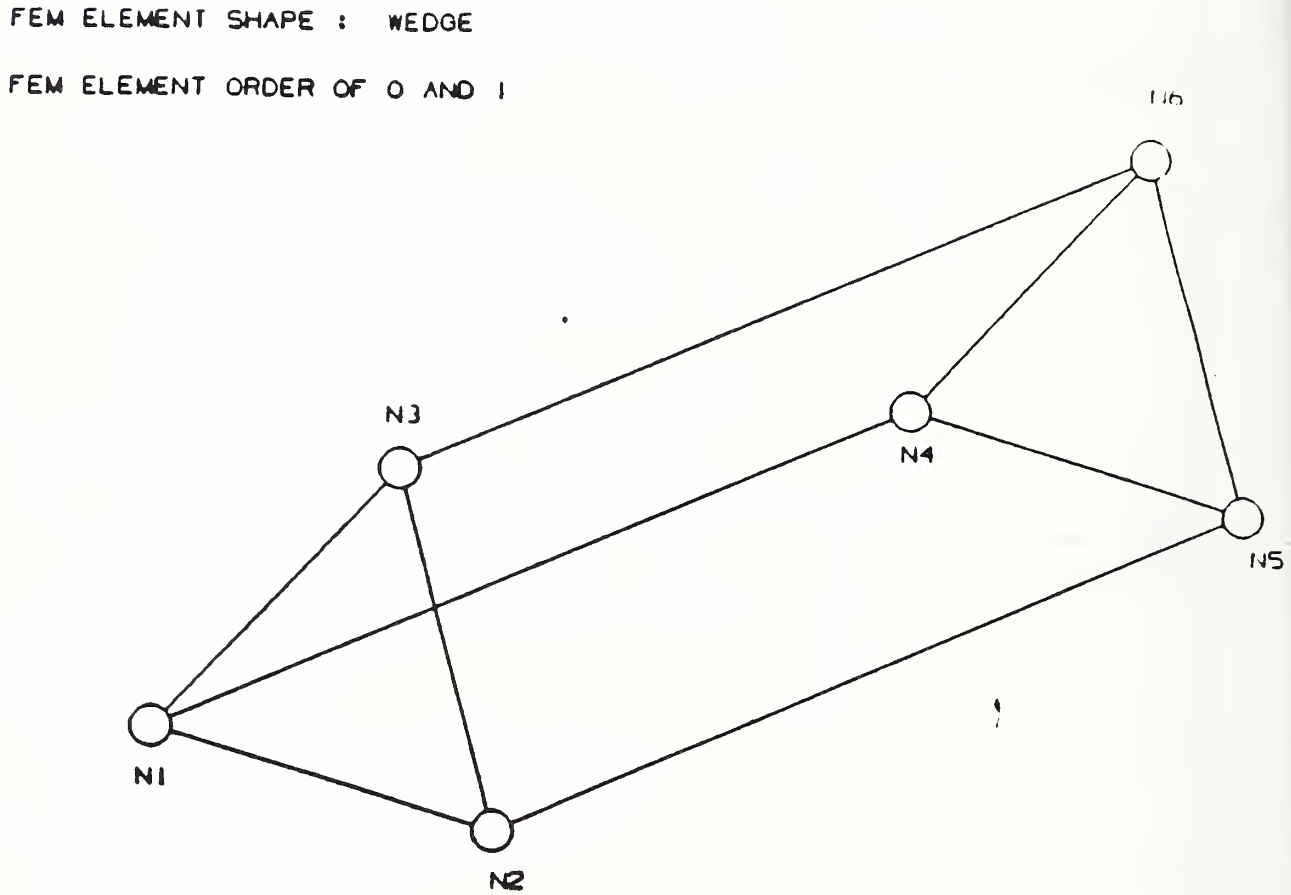


Figure D-133: FEM Wedge Node Connectivity (Order 0 and 1)

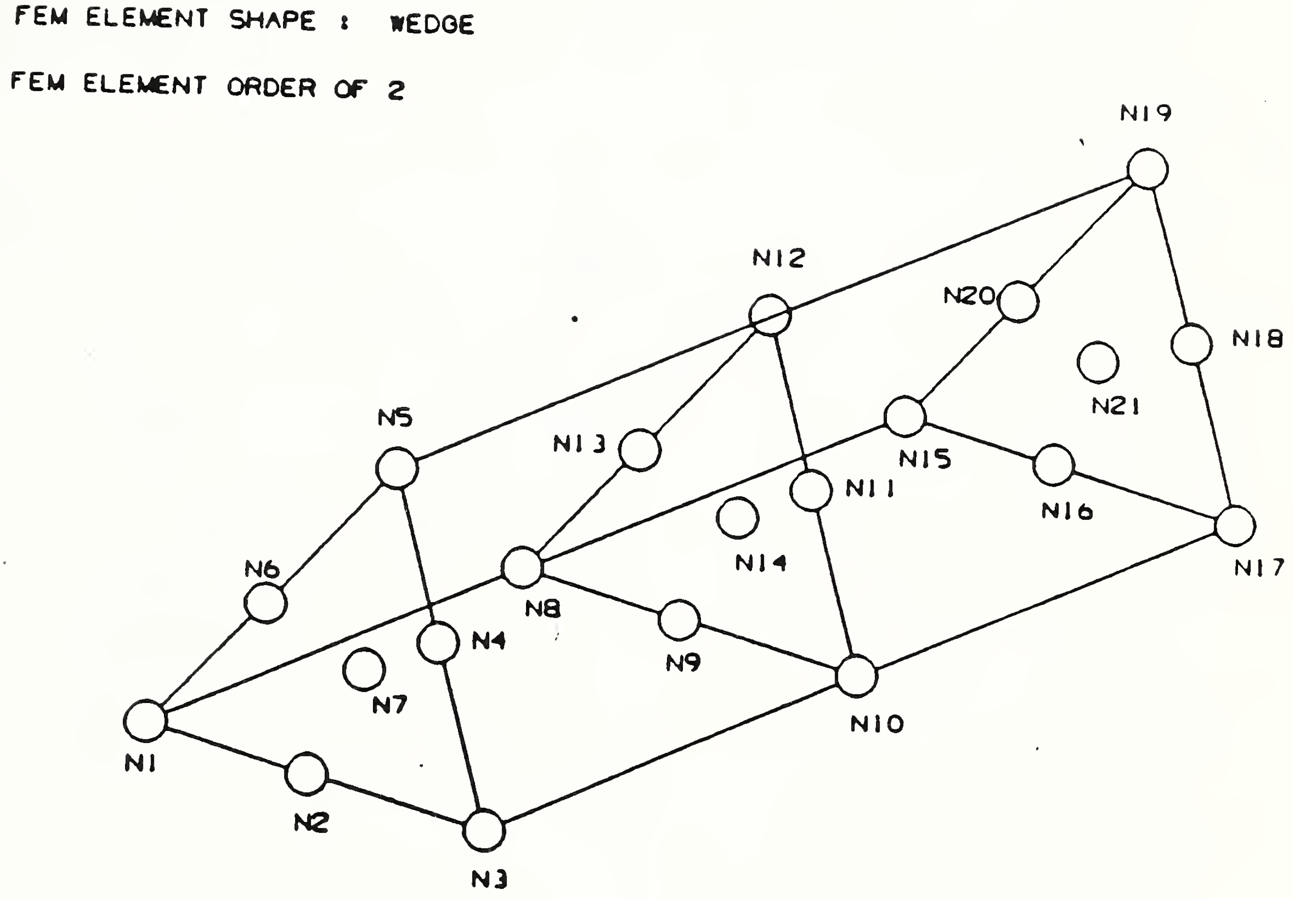


Figure D-134: FEM Wedge Node Connectivity (Order 2)

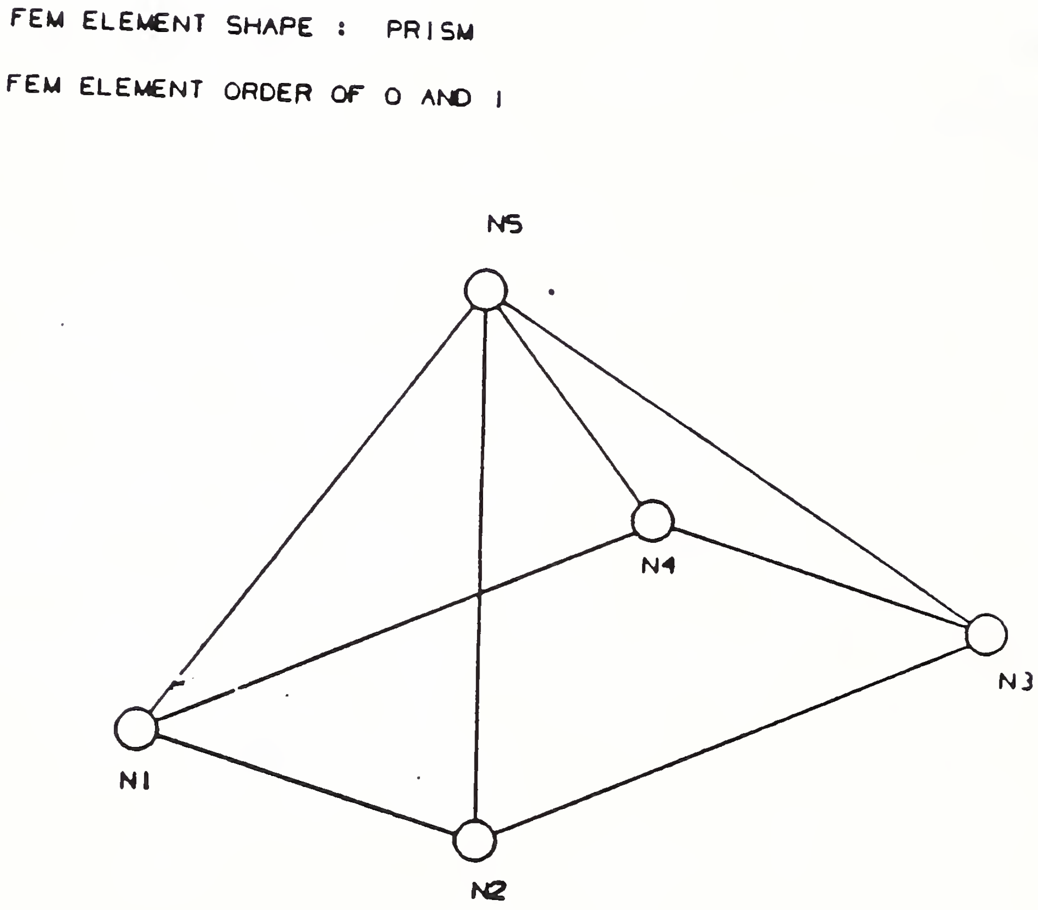


Figure D-135: FEM Prism Node Connectivity (Order 0 and 1)

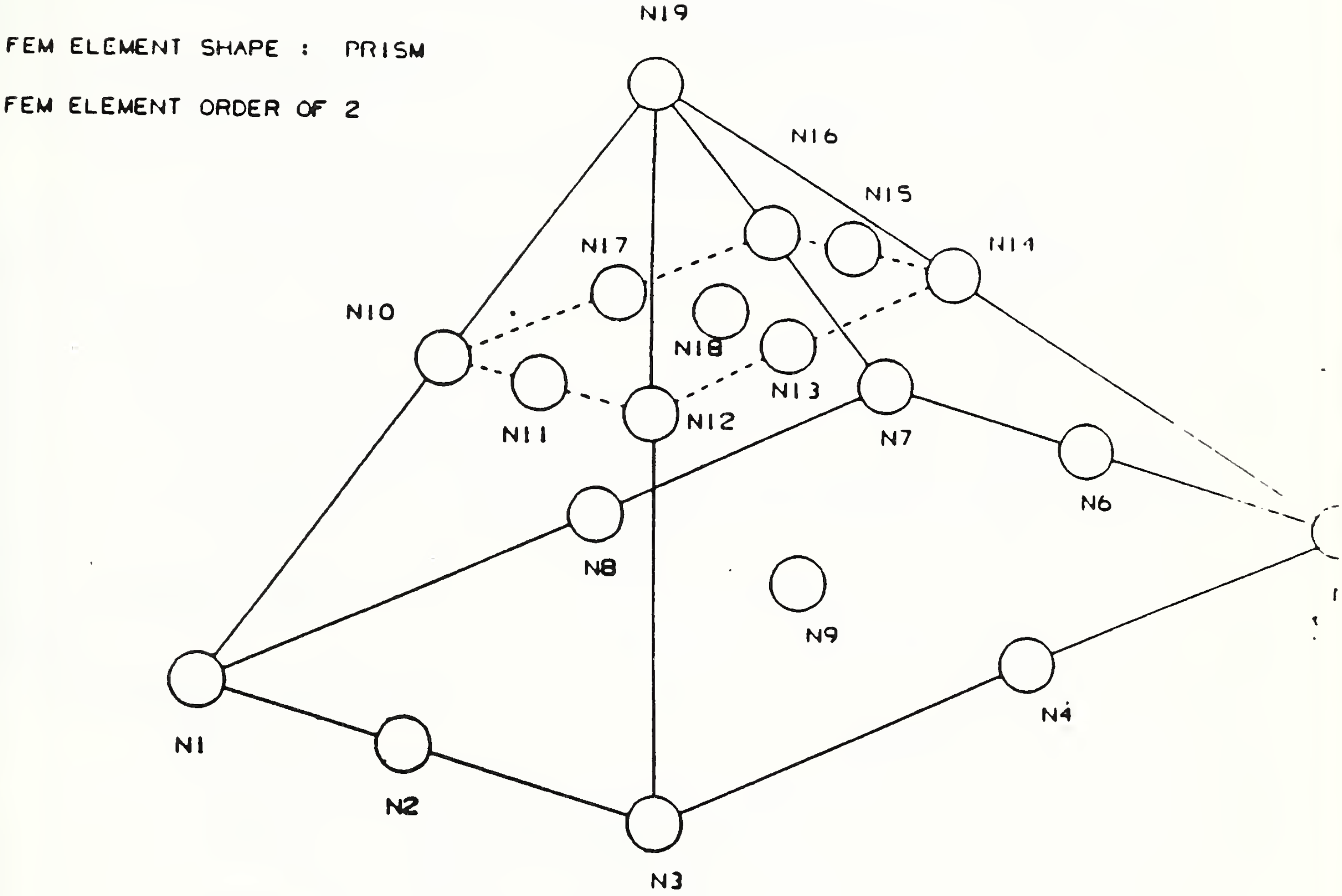


Figure D-136: FEM Prism Node Connectivity (Order 2)

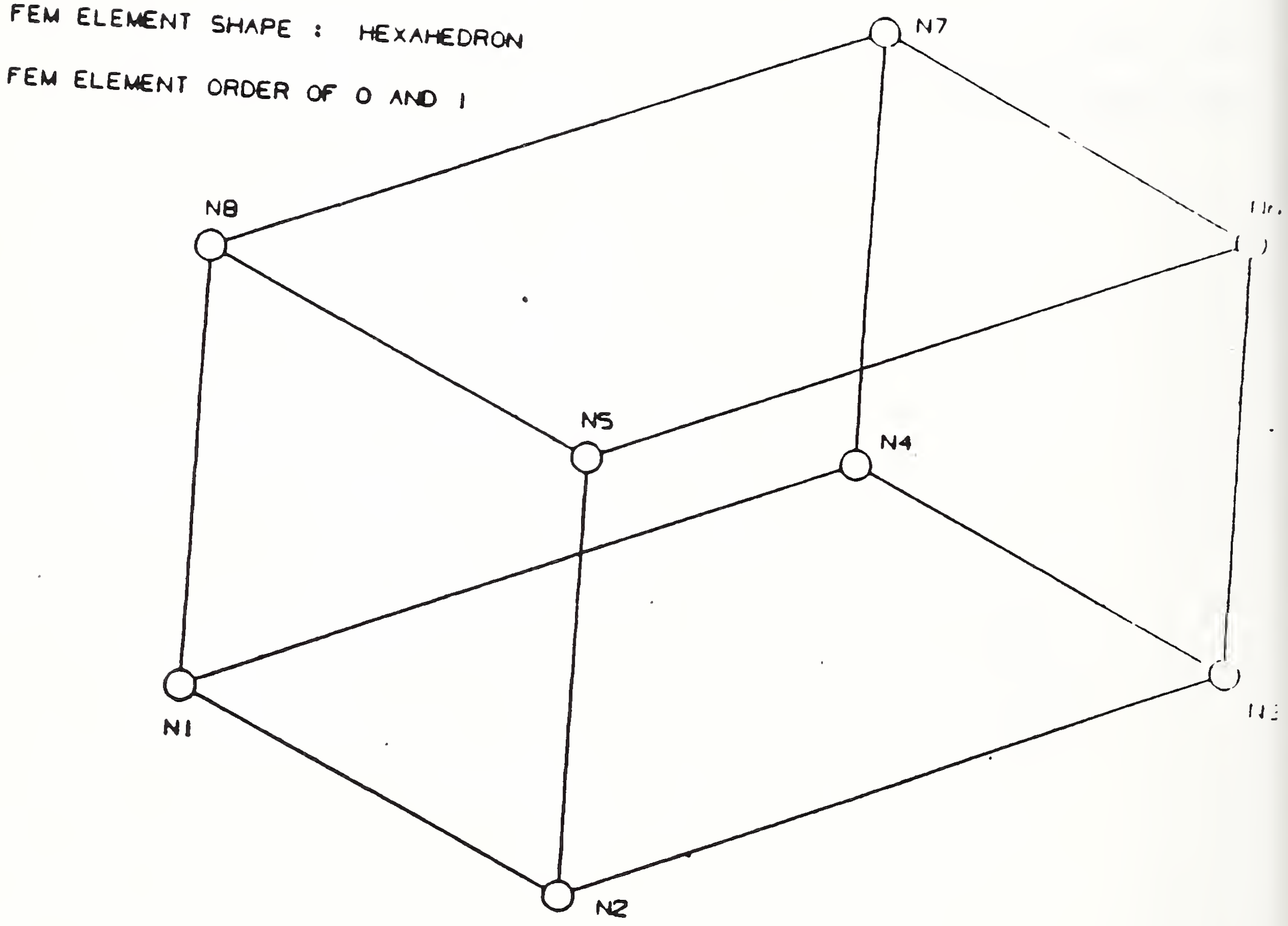


Figure D-137: FEM Hexahedron Node Connectivity (Order 0 and 1)

Entity Name: **FEM GROUP**

Entity Number: **FEM-7**

This entity contains data about a set of FEM entities that, for convenience, can be referred to by referring to one grouped item. Groups of elements, nodes, and environments are currently defined and they may be arbitrarily combined. Thus groups (sets) of nodes and finite elements can be referenced by referring to a single FEM GROUP data entity.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

FEM GROUP NUMBER Data Type: Integer

A number used to identify a FEM GROUP from all others within a FEM. It is only unique within the current FEM. A FEM GROUP can be made up of nodes, elements, or any combination of the two.

Other Attributes:

FEM GROUP NAME Data Type: String

Text that describes the FEM GROUP. For example, possible values of this attribute might be 'red', 'set 1', or 'wing'. It is what ever the analyst chooses as a label for group identification.

Business rules:

EXPRESS specification:

```
ENTITY group;
  group_number   : integer;
  group_name     : OPTIONAL string;
  node_ref       : OPTIONAL EXTERNAL SET [1:#] OF
                   node;
  element_ref    : OPTIONAL EXTERNAL SET [1:#] OF
                   element;
END_ENTITY;
```

Entity Name: **FEM NODE GROUP**

Entity Number: **FEM-8**

The FEM NODE GROUP assigns a FEM NODE to a FEM GROUP.

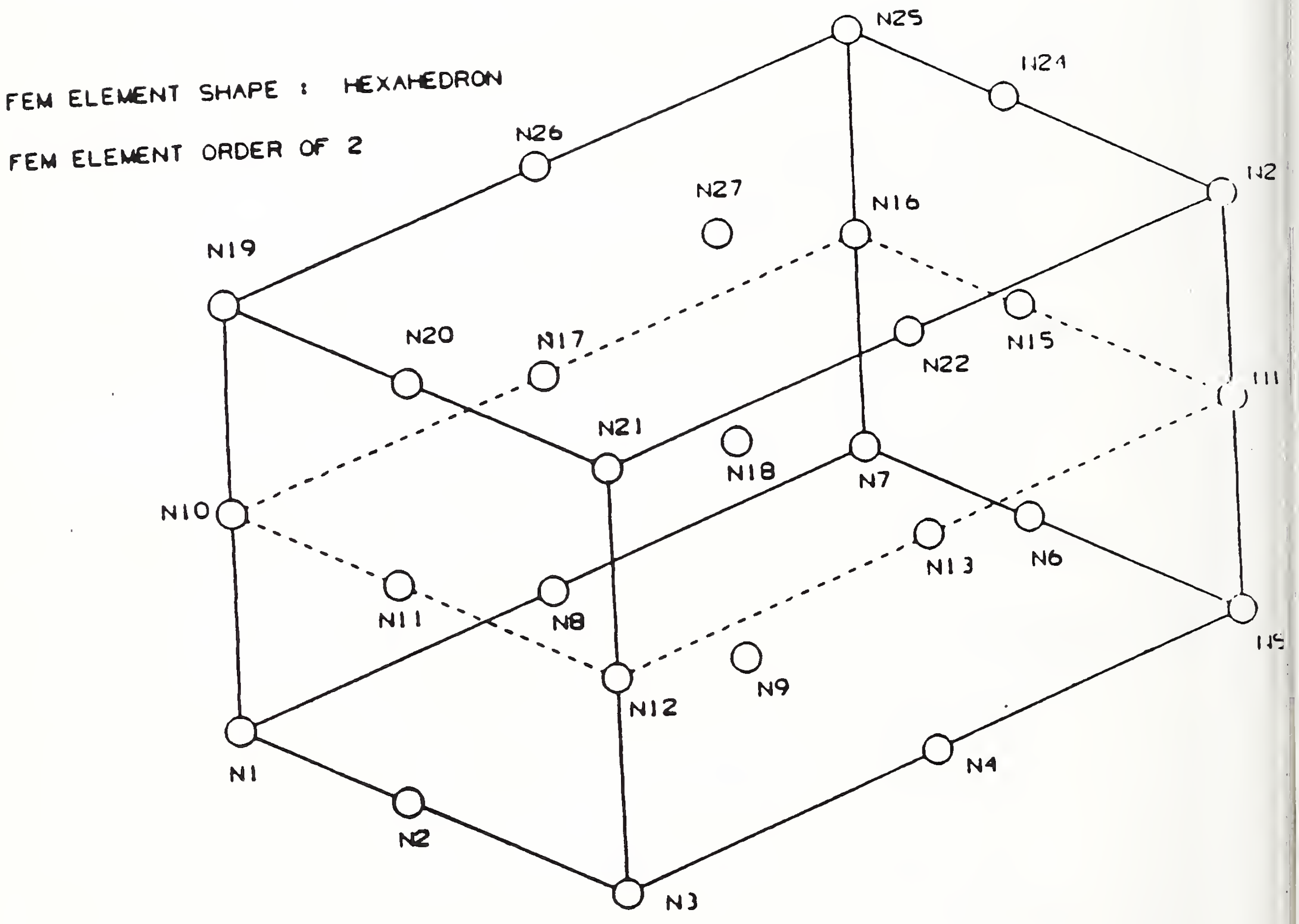


Figure D-138: FEM Hexahedron Node Connectivity (Order 2)

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.

FEM GROUP NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GROUP [Entity FEM-7].

FEM NODE NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM NODE [Entity FEM-16].

Other Attributes:

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1x in this list of FEM NODES represented by this IDEF1X entity was move into the FEM GROUP EXPRESS entity. Therefore see entity FEM GROUP [Entity FEM-7] for details.

Entity Name: **FEM ELEMENT GROUP**

Entity Number: **FEM-9**

The FEM ELEMENT GROUP assigns a FEM ELEMENT to a FEM GROUP.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.

FEM GROUP NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GROUP [Entity FEM-7].

FEM ELEMENT NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM ELEMENT [Entity FEM-17].

Other Attributes:

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1X in this list of FEM ELEMENTs represented by this IDEF1X entity was move into the FEM GROUP EXPRESS entity. Therefore see entity FEM GROUP [Entity FEM-7] for details.

Entity Name: **FEM GEOMETRIC PROPERTY**

Entity Number: **FEM-11**

This entity contains data that describes the physical and geometrical characteristics of a finite element.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.

FEM GEOMETRIC PROPERTY NUMBER Data Type: Integer
A number used to identify a FEM GEOMETRIC PROPERTY from all others within a FEM. It is only unique within the current FEM.

Other Attributes:

FEM GEOMETRIC PROPERTY TYPE Data Type: Enumeration
An enumerated value used to identify the type of FEM GEOMETRIC PROPERTY. The possible types are:

Geometric Type	Description
0	Beam property
1	Point property
2	Shell property
3	Membrane property
4	Solid property

Business rules:

EXPRESS specification:

```
ENTITY geometric_property SUPERTYPE OF (
        point_geometric_property OR
        line_geometric_property OR
        beam_geometric_property OR
        membrane_shell_geometric_property OR
        solid_geometric_property);
  geometric_property_number     : integer;
  property_text : OPTIONAL string;
  integration_order     : OPTIONAL integer;
END_ENTITY;
```

Entity Name: **FEM MATERIAL PROPERTY**

Entity Number: **FEM-12**

This entity contains data that describes a material constitutive matrix for either a FEM ELEMENT or a FEM NODE.

In the general case, the thermoelastic behavior of real materials is relatively complex and is influenced by factors including the material properties, magnitude and type of the loads, temperature, time, the rate of loading or deformation, and the previous history of the material. At any point of the material however, at any given time, temperature, and rate of loading or deformation, the behavior can be characterized by a constitutive law for the material. The constitutive laws EXPRESS the relationships between the material internal forces, or stresses, and material deformations, or strains. At any point within a material, this incremental relationship between stresses and strains is characterized by a set of elastic moduli, thermal expansion coefficients, and structural damping coefficients, which characterize the material behavior. These constants therefore define the material's properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

FEM MATERIAL PROPERTY NUMBER Data Type: Integer

A number used to identify a FEM MATERIAL PROPERTY from all others within a FEM. It is only unique within the current FEM.

Other Attributes:

FEM MATERIAL PROPERTY TYPE Data Type: Enumeration

An enumeration used to identify the type of FEM MATERIAL PROPERTY. The possible types are:

Material Type	Description
0	Composite material property
1	Material table
2	Homogeneous material property

Business rules:

EXPRESS specification:

```
ENTITY material_property SUPERTYPE OF (
        homogeneous_material OR
        composite_material OR
        material_table);
  material_property_number      : integer;
  text  : OPTIONAL string;
END_ENTITY;
```

Entity Name: FEM ELEMENT GEOMETRIC PROPERTY

Entity Number: FEM-13

The FEM ELEMENT GEOMETRIC PROPERTY assigns a FEM GEOMETRIC PROPERTY to a FEM ELEMENT. Use of this data entity for a finite element geometric property implies that they are constant through out the finite element.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

FEM ELEMENT NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM ELEMENT [Entity FEM-17].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1X and the reason is unknown.

Entity Name: **FEM ELEMENT/NODE GEOMETRIC PROPERTY**

Entity Number: **FEM-15**

The FEM ELEMENT/NODE GEOMETRIC PROPERTY assigns a FEM GEOMETRIC PROPERTY to a finite element's node via the FEM CONNECTIVITY data entity and the FEM NODE SEQUENCE NUMBER. Use of this data entity for a finite element's geometric properties implies that they vary through-out the finite element, see rule FEM/R-7. This type of information (if it exists at all) should be handled via an ordered list within the physical file.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM ELEMENT NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM ELEMENT [Entity FEM-17].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

FEM NODE SEQUENCE NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM CONNECTIVITY [Entity FEM-6].

Other Attributes:

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1X and the reason is unknown.

Entity Name: **FEM NODE**

Entity Number: **FEM-16**

A FEM NODE is a place (point) within a FEM that is defined by the analyst. This FEM NODE has attributes defined for it that are specific to the FEM that the FEM NODE is in. Also the FEM NODE is normally a point in space, but it is not always.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM NODE NUMBER Data Type: Integer
A number used to identify a FEM node from all others within a FEM. It is only unique within the current FEM.

Other Attributes:

FEM NODE COORDINATE 1 Data Type: Real

FEM NODE COORDINATE 2 Data Type: Real

FEM NODE COORDINATE 3 Data Type: Real
These attributes contain the location components of a node in the node's FEM NODE DEFINITION COORDINATE SYSTEM with respect to directions 1, 2, and 3, respectively.
The relationship between the defining coordinate system type and the node coordinate values are interpreted in the following table along with the following figure 139.

Coordinate System Type	Node Direction 1	2	3	Figure
Rectangular	X	Y	Z	139
Cylindrical	R	Theta	Z	139
Spherical	R	Theta	Phi	139

FEM NODE DEFINITION COORDINATE SYSTEM NUMBER (FK) The identifier of the FEM COORDINATE SYSTEM used to define the location of the FEM node. See figure 139 for an examples.
See attribute defination in entity FEM COORDINATE SYSTEM [Entity FEM-18].

FEM NODE OUTPUT COORDINATE SYSTEM NUMBER (FK) The identifier of the FEM COORDINATE SYSTEM used to define the output directions of the FEM result data. See figure 139 for an examples.
See attribute defination in entity FEM COORDINATE SYSTEM [Entity FEM-18].

Business rules:

EXPRESS specification:

```
ENTITY node;
  node_number   : integer;
  coordinate_1  : real;
  coordinate_2  : real;
  coordinate_3  : real;
  definition_coordinate_system_ref      : EXTERNAL fem_coordinate_system;
  output_coordinate_system_ref  : EXTERNAL fem_coordinate_system;
END_ENTITY;
```

Entity Name: **FEM ELEMENT**

Entity Number: **FEM-17**

This entity contains data about a finite element. A finite element is the basic building block of a FEM. It defines a mathematical relationship between a FEM's nodes.

In surveying commonly used FEM codes in the United States and Europe, we were able to create a general set of attributes which are flexible enough to be used to defined all of the known element types in use today. These attributes are based on the requirements that the following data attributes be specified in conjunction with each element.

- Shape
- Order
- Element Purpose Descriptor

Using these attributes, see definations below, the essential information regarding an element shape and connectivity information can be prescribed in such a manner as to fully define all element sets in the survey. See reference Rules For Defining Element Topology And Connectivity by Henry Fong and Hayden Hamilton from PDA Engineering dated December 1984 for details.

Primary Key Attributes:

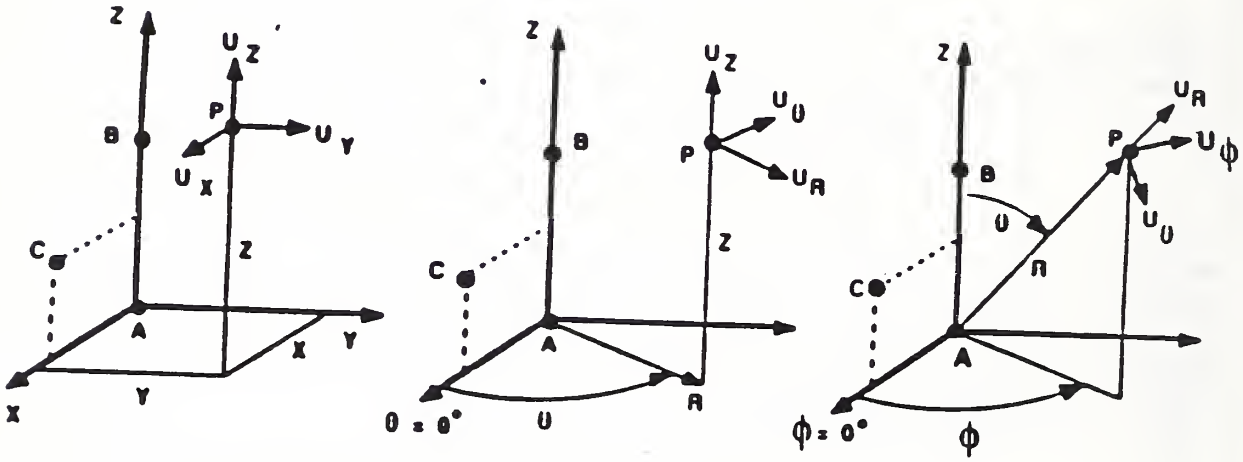
FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM ELEMENT NUMBER Data Type: Integer

A number used to identify a finite element from all others within a FEM. It is only unique within the current FEM.

COORDINATE FRAMES - DEFINITION



RECTANGULAR

X Y Z

CYLINDRICAL

R θ° Z

SPHERICAL

R θ° φ°

Figure D-139: FEM Node Coordinate System

Other Attributes:

FEM ELEMENT GEOMETRIC PROPERTY COORDINATE SYSTEM NUMBER (FK) Data Type: Integer

The identifier of the FEM coordinate system used to define the directions of the geometric properties in the finite element.

FEM ELEMENT LOCAL COORDINATE SYSTEM NUMBER (FK) Data Type: Integer

The identifier of the FEM coordinate system used to define the local directions of the finite element.

FEM ELEMENT MATERIAL PROPERTY COORDINATE SYSTEM NUMBER (FK) Data Type: Integer

The identifier of the FEM coordinate system used to define the directions of the material properties in the finite element.

FEM ELEMENT ORDER NUMBER Data Type: Enumeration

An enumerated value describing the mathematical order of the finite element. The possible values are:

Order number	Order Description
0	constant
1	linear
2	quadrilateral
3	cubic

FEM ELEMENT PURPOSE DESCRIPTOR Data Type: Enumeration

An string enumeration describing the functional purpose of the finite element which represents a user-defined flag (ie enumeration) allowing the differentiation of FEM elements having the same SHAPE and ORDER, but which are different FEM element types appearing in the same FEM data. An example of this situation could be the MSC/NASTRAN CROD and CBAR elements which both have the same SHAPE of line and the same ORDER of linear but behave differently.

The possible strings are:

Purpose number	Descriptor string	Description
P0	mass	A mass is a point element that is a lumped mass element.
L0	rod	A rod is a line element that can only carry axial loads and torsion.
L1	beam	A beam is a line element that can carry loads in all degrees of freedom.
L2	spring	A spring is a line element that is a spring between two nodes in the requested freedom of degree.

L3	damper	A damper is a line element that is a damper between two nodes in the requested freedom of degree.
L4	gr_spring	A gr_spring is a line element that is a spring between a node and ground in the requested freedom of degree.
L5	gr_damper	A gr_damper is a line element that is a damper between a node and ground in the requested freedom of degree.
A0	plate	A plate is a surface element that can carry loads in all degrees of freedom except the moment normal to its surface.
A1	shell	A shell is a surface element that can carry loads in all degrees of freedom.
A2	membrane	A membrane is a surface element that can only carry in-plane loads.
A3	th_shell	A th_shell is a thick shell surface element. It is like the shell surface element, but it takes into account thick shell theory.
A4	tn_shell	A tn_shell is a thin shell surface element. It like the shell surface element, but it takes in account thin shell theory.
A5	pst_shell	A pst_shell is a plane strain shell surface element. It is like the shell surface element, but it takes into account plane strain theory.
A6	pss_shell	A pss_shell is a plane stress shell surface element. It is like the shell surface element, but it takes into account plane stress theory.
A7	ax_shell	A ax_shell is a axisymmetric shell line element. It is like the shell surface element, but it is formed by sweeping the line element around the axis of symmetry.
V0	solid	A solid is a volume element that can carry inplane loads, no moments.
V1	ax_solid	A ax solid is a axisymmetric solid surface element. It is like the solid volume element, but it is formed by sweeping the surface element around the axis of symmetry.

FEM ELEMENT SHAPE NUMBER Data Type: Enumeration

An enumerated value describing the generic shape of the finite element. The possible values are:

Shape number	Generic Shape Description
0	point
1	line
2	triangle
3	quadrilateral
4	tetrahedron
5	wedge
6	prism
7	hexahedron

Business rules:

EXPRESS specification:

```
ENTITY element;
  element_number         : integer;
  geometric_property_coord_sys  : EXTERNAL fem_coordinate_system;
  local_coordinate_system       : EXTERNAL fem_coordinate_system;
  material_property_coord_sys   : EXTERNAL fem_coordinate_system;
  element_purpose_descriptor    : string;
  element_order_number   : element_order;
  element_shape_number   : element_shape;
  node_list      : EXTERNAL ARRAY [1 : maxnodes] OF
                   node;
  geometric_property_type       : enumeration_of_constant_varying;
  geometric_property_ref        : EXTERNAL ARRAY [1 :
                   geom_prop_list_size] OF
                   geometric_property;
WHERE
  maxnodes = derive_maxnodes (element_order,element_shape);
  geom_prop_list_size = derive_geom_prop_list_size (
    geometric_property_type, maxnodes);
  test_geom_prop(enumeration_of_constant_varying;maxnodes;
    node_list;geometric_property_ref) = TRUE;
END_ENTITY;

TYPE element_order = ENUMERATION OF (constant,linear,quadratic,cubic);
END_TYPE;

TYPE element_shape = ENUMERATION OF (
        point,
        line,
        triangle,
        quadrilateral,
        tetrahedron,
        wedge,
        pyramid,
        hexadedron);
END TYPE;

TYPE enumeration_of_constant_varying = ENUMERATION of (
        constant,
        varying);
END_TYPE;

FUNCTION derive_maxnodes (order: element_order, shape:
  element_shape):integer;
```

(* This function is incomplete. It gives an idea of what is required but needs to be extended to include all possible combinations of element_shape and element_order *)

```
LOCAL
  maxnodes : integer;
END_LOCAL;
  IF shape = point THEN
    maxnodes := 1;
  END_IF;
  IF (shape = line AND order < 2) THEN
    maxnodes := 2;
  END_IF;
  IF (shape = line AND order > 1) THEN
    maxnodes := order + 1;
  END_IF;
  IF (shape = triangle AND order < 2) THEN
    maxnodes := 3;
  END_IF;
  IF (shape = triangle AND order = 2) THEN
    maxnodes := 7;
  END_IF;
RETURN (maxnodes);
END_FUNCTION;

FUNCTION derive_geom_prop_list_size
  (attr1:enumeration_of_constant_varying, maxnodes:integer): integer;
LOCAL
```

```
  result : integer;
END_LOCAL;
  IF attr1 = constant THEN
    result := 1
  ELSE
    result := MAXNODES;
  END_IF;
RETURN (result);
END_FUNCTION;

FUNCTION test_geom_prop(attr1:enumeration_of_constant_varying;
  maxnodes:maxnodes;node_list:node_list;
  attr4:geometric_property_ref):LOGICAL;
LOCAL
  result:LOGICAL;
END_LOCAL;
  result := TRUE;
IF attr1 = varying THEN
  REPEAT I:1 TO maxnodes WHILE result = TRUE;
  IF nodes_list(I) <> NULL THEN
    IF attr4(I) = NULL THEN
      result := FALSE;
    ENDIF;
  ENDIF;
ENDIF;
```

Entity Name: **FEM COORDINATE SYSTEM**

Entity Number: **FEM-18**

A reference frame used to define locations of FEM elements, nodes, environments, and results in 2D or 3D space. This data entity contains information about the use (either master or definition) and type (rectangular, polar, etc...) of a coordinate system. Figure 140 illustrates example of coordinate systems.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM COORDINATE SYSTEM NUMBER Data Type: Integer
A number used to identify a coordinate system within a FEM. It is only unique within the current FEM.

:

Other Attributes:

FEM COORDINATE SYSTEM TYPE Data Type: Enumeration

An enumerated value used to identify the type of FEM coordinate system. The possible values are:

Coordinate Type Number	Coordinator Descriptor
0	Right-hand rectangular
1	Right-hand cylindrical
2	Right-hand spherical
3	Left-hand rectangular
4	Left-hand cylindrical
5	Left-hand spherical

FEM COORDINATE SYSTEM USE Data Type: Enumeration

An enumerated value indicating whether the coordinate system is a master or definition coordinate system. The possible values are:

Coordinate Use Number	Coordinator System
0	Master
1	Definition

Business rules:

EXPRESS specification:

```
ENTITY fem_coordinate_system SUPERTYPE OF (
        master_coordinate_system OR
        derived_coordinate_system);
  coordinate_system_number      : integer;
  coordinate_system_type        : coordinate_system_type;
END_ENTITY;

TYPE coordinate_pair = ARRAY [1:2] OF real;
END_TYPE;

TYPE coordinate_triple = ARRAY [1:3] OF real;
END_TYPE;

TYPE  coordinate_system_type = ENUMERATION OF (
        rh_rectangular,
        rh_cylindrical,
        rh_spherical,
:        lh_rectangular,
         lh_cylindrical,
         lh_spherical);
END TYPE;
```

FEM-19 FEM COORDINATE MASTER SYSTEM

A FEM always has a master coordinate system. The origin of the master coordinate system is always at (0,0,0) and the axes are aligned in the primary directions. All other FEM DEFINITION COORDINATE SYSTEMS are derived from the FEM COORDINATE MASTER SYSTEM.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM COORDINATE MASTER SYSTEM NUMBER Data Type: Integer

A number used to identify a master coordinate system within a FEM. There is only one master coordinate system within any one FEM and can be different between FEMs. It is only unique within the current FEM.

Other Attributes:

Business rules:

EXPRESS specification:

```
ENTITY master_coordinate_system SUBTYPE OF (fem_coordinate_system);
END_ENTITY;
```

Entity Name: **FEM COORDINATE TRANSFORM**

Entity Number: **FEM-20**

This data entity defines the transformation matrix used to rotate and shift a FEM MASTER COORDINATE SYSTEM into a FEM DEFINITION COORDINATE SYSTEM.

Primary Key Attributes:

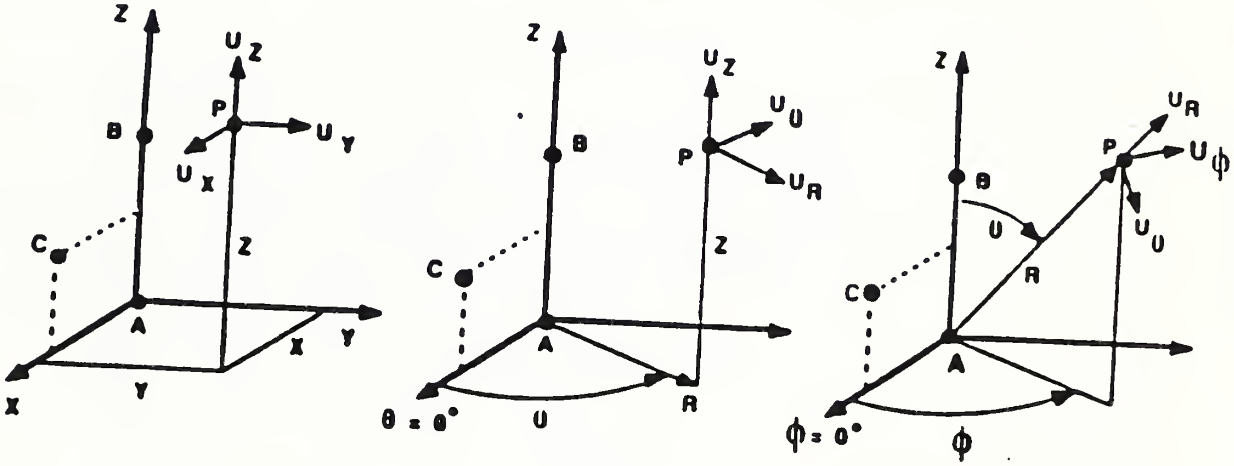
FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM COORDINATE DEFINITION SYSTEM NUMBER (FK) Data Type: String

See attribute defination in entity FEM COORDINATE DEFINITION SYSTEM [Entity FEM-58].

COORDINATE FRAMES - DEFINITION



RECTANGULAR

X Y Z

CYLINDRICAL

R 0° Z

SPHERICAL

R 0° φ°

Figure D-140: FEM Node Coordinate System

FEM COORDINATE MASTER SYSTEM NUMBER (FK) Data Type: String
: See attribute defination in entity FEM COORDINATE MASTER SYSTEM [Entity FEM-58].

Other Attributes:

FEM COORDINATE SYSTEM TRANSFORMATION MATRIX Data Type: String
A matrix to map from the definition coordinate system space to the master coordinate system space. coordinate system space. This matrix is dimensioned 3x4 and has the following form.

		Column			
		1	2	3	4
Row	1	O1	TM11	TM12	TM13
	2	O2	TM21	TM22	TM23
	3	O3	TM31	TM32	TM33

Where O1, O2 and O3 is the location of the origin of the definition coordinate system with respect to directions 1, 2, and 3, respectively in the master coordinate system space. The relationship between the master coordinate system type and the origin coordinate values are interpreted in the following table along with the following figure 139.

Coordinate System Type	Node Direction 1	2	3	Figure
Rectangular	X	Y	Z	139
Cylindrical	R	Theta	Z	139
Spherical	R	Theta	Phi	139

Where TM11 through TM33 are the elements of a 3x3 rotation matrix to transform the coordiantes from this coordinate system to the master cooordinate system.

Business rules:

EXPRESS specification:

The EXPRESS specification is different than the IDEF1x and the reason is that this entity was move into the FEM COORDINATE DEFINITION SYSTEM EXPRESS entity. See entity FEM COOREDINATE DEFINITION SYSTEM [Entity FEM-58] for complete details.

Entity Name: **FEM SHELL GEOMETRIC PROPERTY**

Entity Number: **FEM-24**

This entities data represents a shell finite element's geometric properties. A shell finite element is a special type of 3D finite element.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM SHELL BENDING MATERIAL NUMBER (FK) Data Type: Integer
Material property number for a shell finite element that is used in the element's bending characteristics.
See FEM MATERIAL PROPERTY NUMBER attribute definition in entity FEM MATERIAL PROPERY [Entity FEM-12] for more details.

FEM SHELL MEMBRANE-BENDING MATERIAL NUMBER (FK) Data Type: Integer
Material property number for a shell finite element that is used in the element's membrane-bending characteristics.
See FEM MATERIAL PROPERTY NUMBER attribute definition in entity FEM MATERIAL PROPERY [Entity FEM-12] for more details.

FEM SHELL SECOND MOMENT OF AREA Data Type: Real
The value of the shell finite element's second moment of area, not a second monent of area factor.

FEM SHELL SHEAR MATERIAL NUMBER (FK) Data Type: Integer
Material property number for a shell finite element that is used in the element's shear characteristics.
See FEM MATERIAL PROPERTY NUMBER attribute definition in entity FEM MATERIAL PROPERY [Entity FEM-12] for more details.

FEM SHELL SHEAR THICKNESS Data Type: Real
This attribute contains the value of the shell finite element's thickness for shear. It measured perpendicular to the face of the shell element. This is not a shear thickness factor.

FEM SHELL Z-OFFSET Data Type: Real
This attribute contains the value of the distance by which the shell finite element is offset from its nodes in the local element coordinate system's Z direction.

Business rules:

EXPRESS specification:

The EXPRESS specification is different from the IDEF1X in that this entity was merged into the FEM MEMBRANE/SHELL GEOMETRIC PROPERTY entity. For details see entity FEM MEMBRANE/SHELL GEOMETRIC PROPERTY [Entity FEM-54].

Entity Name: FEM SOLID GEOMETRIC PROPERTY

Entity Number: **FEM-25**

This entities data represents a solid finite element's geometric properties. A solid finite element is a 3D finite element.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM SOLID MATERIAL NUMBER (FK) Data Type: Integer
The identifier of the FEM MATERIAL NUMBER used by a solid finite element.
See FEM MATERIAL PROPERTY NUMBER attribute definition in entity FEM MATERIAL PROPERY [Entity FEM-12] for more details.

Business rules:

EXPRESS specification:

```
ENTITY solid_geometric_property SUBTYPE OF (geometric_property);
  solid_material        : EXTERNAL material_property;
END_ENTITY;
```

Entity Name: FEM POINT GEOMETRIC PROPERTY

Entity Number: **FEM-26**

This entities data represents a FEM point geometric properties. A FEM point is a 0D reference datum.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM POINT MASS MATRIX Data Type: Real
A 6x6 mass half matrix for a point finite element. The matrix is a half matrix because these point mass martix are always symetric and it has the following form.

		Columns					
		1	2	3	4	5	6
Row	1	M1	M2	M3	M4	M5	M6
	2		M7	M8	M9	M10	M11
	3			M12	M13	M14	M15
	4				M16	M17	M18
	5					M19	M20
	6						M21

Business rules:

EXPRESS specification:

```
ENTITY point_geometric_property SUBTYPE OF (geometric_property);
  point_mass_half_6x6_matrix     : ARRAY [1:21] OF real;
END_ENTITY;
```

Entity Name: **FEM BEAM GEOMETRIC PROPERTY**

Entity Number: **FEM-27**

This entities data represents a beam finite element's geometric properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

Business rules:

EXPRESS specification:

```
ENTITY beam_geometric_property SUBTYPE OF (geometric_property);
  pin_array_1_2 : OPTIONAL ARRAY [1 : 2] OF ARRAY
                  [1 : 6] OF integer;
  offset_vector_1_2     : OPTIONAL ARRAY [1 : 2] OF
                  coordinate_triple;
  warping_coefficients_1_2      : OPTIONAL ARRAY [1:2] OF real;
  beam_interval_ref     : EXTERNAL LIST [1 : #] OF
                  beam_interval;
END_ENTITY;
```

Entity Name: **FEM SHELL STRESS RECOVERY COEFFICIENT**

Entity Number: **FEM-28**

These are the values that the analyst wants used for stress recovery with respect to a shell finite element. The coefficient is a z offset from the element's mid-plane in the element's coordinate system.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

FEM SHELL STRESS RECOVERY COEFFICIENT NUMBER Data Type: Integer

Forms, together with the FEM ID and the FEM GEOMETRIC PROPERTY NUMBER the unique identifier for an instance of a FEM SHELL STRESS RECOVERY COEFFICIENT.

Other Attributes:

FEM SHELL STRESS RECOVERY COEFFICIENT Data Type: Real

The value of the distance at which stress will be recovered in the local shell finite element coordinate system's Z direction.

Business rules:

EXPRESS specification:

The EXPRESS specification is different from the IDEF1X in that this entity was merged into the FEM MEMBRANE/SHELL GEOMETRIC PROPERTY entity. For details see entity FEM MEMBRANE/SHELL GEOMETRIC PROPERTY [Entity FEM-54].

Entity Name: **FEM GEOMETRIC PROPERTY TEXT**

Entity Number: **FEM-29**

This data entity supplies ancillary data about a geometric property. For example, a text string describing the geometric properties of a beam could be inserted into this entity. This human readable text string would enable the receiving analyst to determine the cross-sectional shape of an 'I' beam, as this data is not readily available from a beam's cross-sectional and second moments of area. This text may or may not be of value to the receiving analyst or computer.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM GEOMETRIC PROPERTY TEXT Data Type: String

Text that describes the FEM GEOMETRIC PROPERTY and provides supporting information supplied by the analyst.

Business rules:

EXPRESS specification:

The EXPRESS specification is different from the IDEF1X in that this entity was merged into the FEM GEOMETRIC PROPERTY entity. For details see entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Entity Name: **FEM SPRING GEOMETRIC PROPERTY**

Entity Number: **FEM-30**

This data entity representd a FEM spring element's geometric properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

None

Business rules:

None

EXPRESS specification:

```
ENTITY spring_geometric_property;
  property_list : EXTERNAL LIST [1:#] OF
                  spring_property;
END_ENTITY;
```

Entity Name: **FEM DAMPER GEOMETRIC PROPERTY**

Entity Number: **FEM-31**

This data entity represents a FEM damper elment's geometric properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

Business rules:

EXPRESS specification:

```
ENTITY damper_geometric_property;
  property_list : EXTERNAL LIST [1:#] OF
                  damper_property;
END_ENTITY;
```

Entity Name: **FEM HOMOGENEOUS MATERIAL PROPERTY**

Entity Number: **FEM-32**

This data entity represents a material that is homogeneous material. Homogeneous material are: Isotropic, Orthotropic, and Anisotropic materials.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM HOMOGENEOUS MATERIAL TYPE Data Type: Enumeration

An enumeration used to identify the type of FEM HOMOGENEOUS MATERIAL PROPERTY. The possible types are:

Material Type	Description
0	Isotropic material property
1	Orthotropic material property
2	Ansisotropic material property

Business rules:

EXPRESS specification:

```
ENTITY homogeneous_material SUPERTYPE OF (
        isotropic_material OR
        orthotropic_material OR
        anisotropic_material);
        SUBTYPE OF (material_property);
  mass_density  : OPTIONAL real;
  structural_damping_coefficient        : OPTIONAL real;
END_ENTITY;
```

Entity Name: **FEM BEAM PROPERTY DATA**

Entity Number: **FEM-33**

This data entity contains the geometric property values for a particular cross-section instance of a FEM beam element.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM BEAM SECTION NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM BEAM SECTION [Entity FEM-71].

Other Attributes:

FEM BEAM CENTROID OFFSET Data Type: Real

Two coordinate, y and z, referenced to the local element coordiante system, are used to defined the amount by which the section centriod of the beam section is offset from the local element center line (element's neutral axis).

FEM BEAM CROSS SECTION AREA The value of this attribute is the cross-sectional area of a beam finite element.	Data Type: Real
FEM BEAM NON-STRUCTURAL MASS The value of the non-structural mass of the beam finite element for the purpose of applying gravity loading conditions.	Data Type: Real
FEM BEAM NON-STRUCTURAL MASS OFFSET Two coordinates, y, and z, referenced to the local element coordinate system, are used to define the amount by which the non-structural mass is offset from the local element center line (element's offset neutral axis).	Data Type: Real
FEM BEAM POLAR MOMENT OF AREA The value of this attribute contains the beam finite element's polar moment of area.	Data Type: Real
FEM BEAM SECOND MOMENT OF AREAS The values of these three attributes contain the beam finite element's second moment of areas (I11, I22, and I12). The directions are referenced to the local element coordinate system.	Data Type: Real
FEM BEAM SHEAR AREAS The values of these two attributes contain the beam finite element's shear areas (K1, K2). The directions are referenced to the local element coordinate system.	Data Type: Real
FEM BEAM SHEAR CENTER OFFSET Two coordinate, y and z, referenced to the local element coordiante system, are used to defined the amount by which the shear center of the beam section is offset from the local beam center line (element's neutral axis).	Data Type: Real
FEM BEAM SHEAR RELIEF COEFFICIENTS The shear relief coefficients for a beam section.	Data Type: Real

Business rules:

EXPRESS specification:

```
ENTITY beam_property_data SUBTYPE OF (beam_section);
  cross_sectional_area  : real;
  second_moment_of_area_I11     : real;
  second_moment_of_area_I22     : real;
  second_moment_of_area_I12     : real;
  polar_moment  : real;
  shear_area_K1 : real;
  shear_area_K2 : real;
  shear_relief_coefficient_S1   : OPTIONAL real;
  shear_relief_coefficient_S2   : OPTIONAL real;
  non_structural_mass   : OPTIONAL real;
  centroid_offset       : OPTIONAL coordinate_pair;
  shear_centre_offset   : OPTIONAL coordinate_pair;
  non_structural_mass_offset    : OPTIONAL coordinate_pair;
END_ENTITY;
```

Entity Name: **FEM BEAM STRESS RECOVERY COEFFICIENT**

Entity Number: **FEM-34**

This data entity contains y and z coordinate offset values (referenced to the local beam element coordinate system) that are to be used in the stress recovery.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute defination in entity FEM [Entity FEM-1].

FEM BEAM SECTION NUMBER (FK) Data Type: Integer

See attribute defination in entity FEM BEAM SECTION [Entity FEM-71].

FEM BEAM STRESS RECOVERY COEFFICIENT NUMBER Data Type: Integer

This attribute represents a location in the sequence of a beam finite element's stress recovery coefficients.

Other Attributes:

FEM BEAM LOCAL Y COORDINATE Data Type: Real

The value of this attribute contains the Y coordinate of the point at which the stresses are to be calculated for output. This value is referenced to the local beam finite element coordinate system.

FEM BEAM LOCAL Z COORDINATE Data Type: Real

The value of this attribute contains the Z coordinate of the point at which the stresses are to be calculated for output. This value is referenced to the local beam finite element coordinate system.

Business Rules:

EXPRESS Specification:

This express entity is different than the IDEF1X entity in that it was merge into the FEM BEAM SECTION entity [Entity FEM-71].

Entity Name: **FEM COMPOSITE MATERIAL PROPERTY**

Entity Number: **FEM-35**

This data entity represents the concept that some materials are defined as the combination of several materials combined together to form a composite material. Presently, there are three different types of composite materials defined with the FEM IDEF1X data model. These are MIXTURE, HALPIN-TSAI, and LAMINATE. Each type of composite material represents a different computational theory with which to calculate the finite element's constitutive properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM COMPOSITE MATERIAL PROPERTY TYPE (FK) Data Type: Integer
An enumerated value used to identify the type of FEM COMPOSITE MATERIAL PROPERTY. The possible values are:

Composite Material Type	Kind of Composite Material Property
0	Halpin-Tsai
1	Laminate
2	Mixture

Business Rules:

EXPRESS Specification:

```
ENTITY composite_material SUPERTYPE OF (
        halpin-tsai_material OR
        laminate_composite_material OR
        mixture_composite_material)
        SUBTYPE OF (material_property);
END_ENTITY;
```

Entity Name: **FEM ISOTROPIC MATERIAL PROPERTY**

Entity Number: **FEM-36**

This data entity represents a material whose constitutive properties are the same in all directions.

Primary Key Attributes:

FEM ID (FK) This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].	Data Type: String
FEM MATERIAL PROPERTY NUMBER (FK) See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].	Data Type: Integer

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY isotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property      : OPTIONAL EXTERNAL
                isotropic_thermal_property;
  structural_property   : OPTIONAL EXTERNAL
                isotropic_structural_property;
  expansion_property    : OPTIONAL EXTERNAL
                isotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)
  AND (expansion_property = NULL));
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC MATERIAL PROPERTY**

Entity Number: **FEM-37**

This data entity represents a material whose constitutive properties are non-orthogonal.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property       : OPTIONAL EXTERNAL
                 anisotropic_thermal_property;
  structural_proerty     : OPTIONAL EXTERNAL
                 anisotropic_structural_property;
  expansion_property     : OPTIONAL EXTERNAL
                 anisotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)
  AND (expansion_property = NULL));
END_ENTITY;
```

Entity Name: **FEM ISOTROPIC STRUCTURAL MATERIAL PROPERTY**

Entity Number: **FEM-38**

This data entity contains isotropic structural material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ISOTROPIC STRUCTURAL POISSON'S RATIO Data Type: Real
Poisson's ratio represents the ratio of lateral contraction to longitudinal extension in the simple uniaxial tension test.

FEM ISOTROPIC STRUCTURAL SHEAR MODULUS Data Type: Real
The shear modulus, G, of an isotropic material represents the ratio of shearing stress to shearing strain. Whatever the stress system may be, the ratio is measured by relating the shearing strain for any pair of rectangular axes and the shearing stress on a pair of planes orthogonal to these axes.

FEM ISOTROPIC STRUCTURAL YOUNG'S MODULUS Data Type: Real
Young's Modulus for an isotropic elastic material is the ratio of stress to strain observed in a simple uniaxial tension test with traction free lateral surfaces. Young's modulus is denoted by E.

Business Rules:

EXPRESS Specification:

```
ENTITY isotropic_structural_property;
  Youngs_modulus        : real;
  poisson's_ratio       : real;
  shear_modulus : real;
END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC MATERIAL PROPERTY**

Entity Number: **FEM-39**

This data entity represents a material whose constitutive properties are orthogonal, but not the same in the three principal directions.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property       : OPTIONAL EXTERNAL
                 orthotropic_thermal_property;
  structural_proerty     : OPTIONAL EXTERNAL
                 orthotropic_structural_property;
  expansion_property     : OPTIONAL EXTERNAL
                 orthotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)
  AND (expansion_property = NULL));
END_ENTITY;
```

Entity Name: **FEM ISOTROPIC THERMAL MATERIAL PROPERTY**

Entity Number: **FEM-40**

This data entity contains isotropic thermal material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ISOTROPIC HEAT CAPACITY Data Type: Real

The heat capacity of a material is the total quantity of heat required to produce a one unit change in temperature for a unit mass of the material at constant pressure. The heat capacity is a scalar quantity independent of direction for all material types, whether isotropic, orthotropic, or anisotropic.

FEM ISOTROPIC THERMAL CONDUCTIVITY COEFFICIENT Data Type: Real
Fourier's law of heat conduction relates the rate of heat flow through any surface area within the material to the temperature gradient normal to that surface. The constant of proportionality is the material thermal conductivity, which for isotropic materials is independent of direction.

FEM FEM ISOTROPIC THERMAL CONVECTIVE FILM COEFFICIENT Data Type: Real
The convective film coefficient is the constant of proportionality that relates the magnitude of the heat flux normal to an external material boundary to the temperature difference occurring across the boundary.

Business Rules:

EXPRESS Specification:

```
ENTITY isotropic_thermal_property;
  thermal_conductivity_coefficient      : real;
  specific_heat_capacity          : real;
  convective_film_coefficient     : real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC THERMAL 3D MATERIAL PROPERTY**

Entity Number: **FEM-41**

This data entity contains anisotropic thermal 3D material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC THERMAL 3D CONDUCTIVITY MATRIX Data Type: Integer
For anisotropic materials the proportionality constant that relates the rate of heat flow through any surface area within the material to the temperature gradient is dependent

on material directions. For three principal material directions, these constants of proportionality are the material thermal conductivities, denoted as KXX, KYY, and KZZ.

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_thermal_3d_property SUBTYPE OF (
        anisotropic_thermal_property);
  thermal_conductivity  : ARRAY [1:3] OF real;
END_ENTITY;
```

Entity Name: **FEM MIXTURE COMPOSITE MATERIAL PROPERTY**

Entity Number: **FEM-42**

This data entity indicates a mixture composite material.

Primary Key Attributes:

FEM ID (FK) This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].	Data Type: String
FEM MATERIAL PROPERTY NUMBER (FK) See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].	Data Type: Integer

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY mixture_composite_material SUBTYPE OF (
        composite_material);
  mixture_material      : EXTERNAL LIST [1:#] OF
                    mixture_material_property;
WHERE
```

```
  sum(mixture_material.mixture_fraction) = 1.0;
END_ENTITY;
```

Entity Name: **FEM HALPIN-TSAI MATERIAL PROPERTY**

Entity Number: **FEM-43**

This data entity represents a composite material whose material properties are determined using the Halpin-Tsai empirical material model. An example of a Halpin-Tsia composite material could be unidirectional T50 fiber in an isotropic carbon matrix.

This is done by creating the composite properties of a two-phase composite (matrix and fiber) which is transverly isotropic using the Halpin-Tsai empirical model equations. The composite thermal expansion properties are computed using Levin's exact solution for two-phase composites.

The following Halpin-Tsia equations are used to calculate the composite engineering constants:

$$E_l = C(E_1) * (V_f E^{f/l} + V_m E_l^m)$$

$$v_{lt} = C(v_{lt}) * (V_f v^{f/lt} + V_m v_{lt}^m)$$

$$n_1 = (E_t^f - E_t^m)/(E_t^f + E_t^m C(E_t))$$

$$n_2 = (v_{tt}^f - v_{tt}^m)/(v_{tt}^f + v_{tt}^m C(v_{tt}))$$

$$n_3 = (G_{lt}^f - G_{lt}^m)/(G_{lt}^f + G_{lt}^m C(G_{lt}))$$

$$E_t = E_t^m * (1.0 + V_f n_1 C(E_t))/(1.0 - V_f)$$

$$v_{tt} = v_{tt}^m * (1.0 + V_f n_2 C(v_{tt}))/(1.0 - V_f)$$

$$G_{lt} = G_{lt}^m * (1.0 + V_f n_3 C(G_{lt}))/(1.0 - V_f)$$

Where:
V_f is the volume factor for the fiber.
V_m is the volume factor for the matrix.
E_l is the Young's modulus in the longitudinal direction.
E_t is the Young's modulus in the transverse direction.
v_{lt} is the Poisson's ratio in the longitudinal-transverse direction.
G_{lt} is the Shear modulus in the longitudinal-transverse direction.
$C(E_l), C(v_{lt}), C(E_t), C(v_{tt})$ and $C(G_{lt}$ are Halpin-Tsai coefficients for their respective engineering material constants.
The subscript of f is for fiber.
The subscript of m is for matrix.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM HALPIN-TSAI COEFFICIENTS Data Type: Real array

The Halpin-Tsai coefficients are $C(E_l), C(v_{lt}), C(E_t), C(v_{tt})$ and $C(G_{lt})$ for their respective engineering constants.

FEM HALPIN-TSAI FIBER MATERIAL NUMBER Data Type: Integer

The FEM HALPIN-TSIA FIBER MATERIAL NUMBER is a pointer to a FEM MATERIAL PROPERTY entities that will be used as the fiber for this Halpin-Tsai material.

FEM HALPIN-TSAI FIBER VOLUME FRACTION Data Type: Real

The fiber volume fraction is one of two fractional numbers that sum to 1 describing the proportions of the mix of the fiber and matrix materials in a Halpin-Tsai material.

FEM HALPIN-TSAI MATRIX MATERIAL NUMBER Data Type: Integer

The FEM HALPIN-TSIA MATRIX MATERIAL NUMBER is a pointer to a FEM MATERIAL PROPERTY entities that will be used as the matrix for this Halpin-Tsai material.

FEM HALPIN-TSAI MATRIX VOLUME FRACTION Data Type: Real

The matrix volume fraction is one of two fractional numbers that sum to 1 describing the proportions of the mix of the fiber and matrix materials in a Halpin-Tsai material.

Business Rules:

EXPRESS Specification:

```
ENTITY halpin-tsai-material SUBTYPE OF (composite_material);
  fiber_volume_fraction : real;
  matrix_volume_fraction        : real;
  coefficients  : LIST [1:#] OF real;
  fiber_material         : EXTERNAL material_property;
  matrix_material        : EXTERNAL material_property;
WHERE
  fiber_volume_fraction+matrix_volume_fraction = 1.0;
END_ENTITY;
```

Entity Name: **FEM LAMINATE COMPOSITE MATERIAL PROPERTY**

Entity Number: **FEM-44**

This data entity represents a composite material whose material properties are determined by laminates (layers) stacked together. Each laminate is a single material of a constant thickness.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM LAMINATE Z OFFSET Data Type: Real
The laminate Z-offset is the distance from nodal plane (also mid-plane of laminate) to the top of the laminate.

Business Rules:

EXPRESS Specification:

```
ENTITY laminate_composite_material SUBTYPE OF (composite_material);
  laminate_z_offset      : real;
  laminate_material      : EXTERNAL LIST [1:#] OF
                       laminate_material_property;
END_ENTITY;
```

Entity Name: **FEM MIXTURE MATERIAL PROPERTY**

Entity Number: **FEM-45**

This data entity represents a composite material whose material properties are determined by the rule of mixtures.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

FEM MIXTURE VOLUME SEQUENCE NUMBER Data Type: Integer

This key attribute is an artifact of the IDEF1X modeling process. Rather than have a real data type for a key attribute, this attribute was artificially created to define a unique key. An ordered list will suffice in EXPRESS.

Other Attributes:

FEM MIXTURE VOLUME FRACTION Data Type: Real

Three mixture material attributes are defined here. These attributes are related. For each mixture material, a volume material number will exist and point to material properties for that type of material. For each volume material number, there will exist a volume fraction which is a fractional amount of this type of material in this element. The sum of the volume fractions for all volume material numbers in this element must be 1. There will also be a corresponding volume orientation for each volume material number which defines the direction of the volume material with respect to the local element coordinate system.

FEM MIXTURE VOLUME MATERIAL NUMBER (FK) Data Type: Integer

Three mixture material attributes are defined here. These attributes are related. For each mixture material, a volume material number will exist and point to material properties for that type of material. For each volume material number, there will exist a volume fraction which is a fractional amount of this type of material in this element. The sum of the volume fractions for all volume material numbers in this element must be 1. There will also be a corresponding volume orientation for each volume material number which defines the direction of the volume material with respect to the local element coordinate system.

FEM MIXTURE VOLUME ORIENTATION Data Type: Real

Three mixture material attributes are defined here. These attributes are related. For each mixture material, a volume material number will exist and point to material properties for that type of material. For each volume material number, there will exist a volume fraction which is a fractional amount of this type of material in this element. The sum of the volume fractions for all volume material numbers in this element must be 1. There will also be a corresponding volume orientation for each volume material number which defines the direction of the volume material with respect to the local element coordinate system.

Business Rules:

EXPRESS Specification:

```
ENTITY mixture_material_property;
  volume_fraction        : real;
  volume_orientation     : real;
  volume_material        : EXTERNAL material_property;
WHERE;
  0.0 < volume_fraction < 1.0;
END_ENTITY;
```

Entity Name: **FEM LAMINATE MATERIAL PROPERTY**

Entity Number: **FEM-46**

This data entity represents data about a laminate's material properties. Each laminate is define by a material, in-plane orientation, and thickness.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

FEM LAMINATE PLY SEQUENCE NUMBER Data Type: Integer
The laminate ply sequence number is a construct that relates the other attributes of this type to their position within a finite element. By convention plies will be numbered sequentially from the bottom (as determined by the local element coordinate system) to the top of the element.

Other Attributes:

FEM LAMINATE PLY MATERIAL NUMBER (FK) Data Type: Integer
The laminate ply material number points to the material property for the ply.

FEM LAMINATE PLY ORIENTATION Data Type: Real
The laminate ply orientation is the in-plane angle between the material's major direction and the local element coordinate system.

FEM LAMINATE PLY THICKNESS Data Type: Real
The laminate ply thickness is the thickness (in appropriate units) for the ply.

Business Rules:

EXPRESS Specification:

```
ENTITY laminate_material_property;
  ply_thickness : real;
  ply_orientation       : real;
  ply_material  : EXTERNAL material_property;
END ENTITY;
```

Entity Name: **FEM ORTHOTROPIC STRUCTURAL 2D MATERIAL PROPERTY**

Entity Number: **FEM-47**

This data entity contains the additional orthotropic structural 2D material information that is not in entity FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY [Entity FEM-76]. Please note that currently there is no information in this entity and it was added for clarity purposes.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_structural_2d_property SUBTYPE OF
  (orthotropic_STRUCTURAL_property);
END_ENTITY;
```

Entity Name: FEM ORTHOTROPIC THERMAL 2D MATERIAL PROPERTY

Entity Number: FEM-48

This data entity contains orthotropic thermal 2D material information that is not in entity FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY [Entity FEM-77]. Please note that currently there is no information in this entity and it was added for clarity purposes.

Primary Key Attributes:

FEM ID (FK)	Data Type: String
This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].	
FEM MATERIAL PROPERTY NUMBER (FK)	Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].	

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_thermal_2d_property SUBTYPE OF
  (orthotropic_thermal_property);
END_ENTITY;
```

Entity Name: FEM ORTHOTROPIC STRUCTURAL 3D MATERIAL PROPERTY

Entity Number: FEM-49

This data entity contains orthotropic structural 3D material information that is not in entity FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY [Entity FEM-76].

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE POISSON'S RATIOS Data Type: Real

V23 and V31 are the out-of-plane Poisson's ratios for a orthotropic material.

FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE SHEAR MODULI Data Type: Real

G23 and G31 are the out-of-plane shear moduli for an orthotropic material.

FEM ORTHOTROPIC STRUCTURAL OUT-OF-PLANE YOUNG'S MODULUS Data Type: Real

E33 is the out-of-plane Young's modulus for an orthotropic material.

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_structural_3d_property SUBTYPE OF
  (orthotropic_structural_property);
  out-of-plane_shear_moduli      : ARRAY [1:2] OF real;
  out-of-plane_poissons_ratio    : ARRAY [1:2] OF real;
  out-of-plane_youngs_modulus    : real;
END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC THERMAL 3D MATERIAL PROPERTY**

Entity Number: **FEM-50**

This data entity contains orthotropic thermal 3D material information that is not in entity FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY [Entity FEM-77].

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC THERMAL OUT-OF-PLANE THERMAL CONDUCTIVITY Data Type: Real
K3 is the out-of-plane thermal conductivity for an orthotropic material.

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_thermal_3d_property SUBTYPE OF
  (orthotropic_thermal_property);
  out-of-plane_thermal_conductivity     : real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC STRUCTURAL 2D MATERIAL PROPERTY**

Entity Number: **FEM-51**

This data entity contains anisotropic structural 2D material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC STRUCTURAL 2D MATERIAL MATRIX Data Type: Real

The 3x3 symmetric half matrix of elastic coefficients for a 2D anisotropic material relates stress to strain by the relationship. This matrix is symmetric and therefore, only the upper diagonal portion is needed to be defined. Thus there are only six matrix elements required and are as follows:

		Column 1	Column 2	Column 3
Row	1	A1	A2	A3
	2		A4	A5
	3			A6

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_structural_2d_property SUBTYPE OF
  (anisotropic_structural_property);
  material_elasticity_half_3x3_matrix   : ARRAY [1:6] OF real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC STRUCTURAL 3D MATERIAL PROPERTY**

Entity Number: **FEM-52**

This data entity contains anisotropic structural 3D material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC STRUCTURAL 3D MATERIAL MATRIX

Data Type: Real

The 6x6 matrix of elastic coefficients for a 3D anisotropic material relates stress to strain by the relationship This matrix is symmetric and therefore, only the upper diagonal portion is needed to be defined. Thus there are only twenty-one matrix elements required.

		Columns					
		1	2	3	4	5	6
Row	1	A1	A2	A3	A4	A5	A6
	2		A7	A8	A9	A10	A11
	3			A12	A13	A14	A15
	4				A16	A17	A18
	5					A19	A20
	6						A21

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_structural_3d_property SUBTYPE OF
  (anisotropic_structural_property);
  material_elasticity_half_6x6_matrix   : ARRAY [1:21] OF real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC THERMAL 2D MATERIAL PROPERTY**

Entity Number: **FEM-53**

This data entity contains anisotropic thermal 2D material information.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC THERMAL 2D CONDUCTIVITY MATRIX

Data Type: Real

Business Rules:

EXPRESS Specification:

```
ENTITY membrane_shell_geometric_property SUBTYPE OF
  (geometric_property);
  membrane_material     : EXTERNAL material_property;
  thickness     : real;
  nonstructural_mass    : real;
  bending_material      : OPTIONAL EXTERNAL material_property;
  shear_material        : OPTIONAL EXTERNAL material_property;
  membrane-bending_material     : OPTIONAL EXTERNAL material_property;
  second_moment_of_area : OPTIONAL real;
  shear_thickness       : OPTIONAL real;
  z_offset      : OPTIONAL real;
  stress_recovery_coefficient   : OPTIONAL SET OF real;
END_ENTITY;
```

Entity Name: **FEM INTEGRATION ORDER**

Entity Number: **FEM-55**

This data entity contains the integration order of a finite element and is found to be independent of geometric property type. Integration order was found in a simple search to be needed by beam elements, solids and shell.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM INTEGRATION ORDER Data Type: Integer
A number used to identify the order of integration to be used. The following is a list of the currently know integration order values.

Integration Order	Descriptor
0	Constant
1	First order
2	Second order
3	Third order
4	Fourth order

Business Rules:

EXPRESS Specification:

This express entity is different from the IDEF1X entity in that it was merged into the FEM GEOMETRIC PROPERTY express entity. For details see entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Entity Name: **FEM APPROVAL**

Entity Number: **FEM-56**

This data entity contains information about who and when a finite element model was approved.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

APPROVAL ID (FK) Data Type: String
The APPROVAL ID in conjunction with the FEM ID uniquely identifies a who approved the Finite Element Model.
See attribute definition in entity APPROVAL [Entity RA-1] in PSCM data model.

FEM APPROVAL SEQUENCE NUMBER Data Type: Integer
The FEM APPROVAL SEQUENCE NUMBER indentfies the position or occurence of an APPROVAL in a list of APPROVAL for a FEM. The position may or may not be very important because it is usually organizationly dependented.

Other Attributes:

None

Business Rules:

EXPRESS Specification:

This express entity is different from the IDEF1X entity in that it was merged into the FEM express entity. For details see entity FEM [Entity FEM-1].
The express is current incorrect for this.

Entity Name: **FEM COORDINATE DEFINITION SYSTEM**

Entity Number: **FEM-58**

This data entity contains information about the coordinate system used to define the position and/or direction of some other data element.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM COORDINATE DEFINITION SYSTEM NUMBER Data Type: Integer
A number used to identify a definition coordinate system within a FEM. It is only unique within the current FEM. For additional information, see entity FEM COORDINATE SYSTEM [Entity FEM-18].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY derived_coordinate_system SUBTYPE OF (fem_coordinate_system);
  transformation_matrix : ARRAY [1:4] OF ARRAY [1:3] OF real;
  reference_coordinate_system   : EXTERNAL master_coordinate_system;
END_ENTITY;
```

Entity Name: **FEM STRUCTURAL MASS DENSITY**

Entity Number: **FEM-59**

This data entity contains information about a material's mass density.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM STRUCTURAL MASS DENSITY VALUE Data Type: Real
The mass density of a material is its mass per unit volume.

Business Rules:

EXPRESS Specification:

This express entity is different from the IDEF1X entity in that it was merged into the FEM HOMOGENEOUS MATERIAL PROPERTY express entity. For details see entity FEM HOMOGENEOUS MATERIAL PROPERTY [Entity FEM-32].

Entity Name: **FEM STRUCTURAL DAMPING COEFFICIENT**

Entity Number: **FEM-60**

This data entity contains information about a material's damping coefficient.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM STRUCTURAL DAMPING COEFFICIENT VALUE Data Type: Real
The structural damping coefficient is the measure of the percentage dissipation of energy as the material undergoes elastic deformations. Stress = k*force.

Business Rules:

EXPRESS Specification:

This express entity is different from the IDEF1X entity in that it was merged into the FEM HOMOGENEOUS MATERIAL PROPERTY express entity. For details see entity FEM HOMOGENEOUS MATERIAL PROPERTY [Entity FEM-32].

Entity Name: FEM MATERIAL TABLE

Entity Number: FEM-61

This data entity represents the concept that a material's properties sometime vary with respect to independent variables. This data entity implements this concept in a very general manner. Typically, in structural mechanics problems where there is a wide variation of temperature among a FEM's nodes, material property data is input as a tabular set of information with the independent variable being temperature and the dependent variables being the material's Young's Modulus, Poisson's Ratio, etc... These data are interpolated with respect to temperature to determine what material property value to use.

Because this entity (along with its companion entity, FEM MATERIAL TABLE INSTANCE) recursively points to a FEM MATERIAL PROPERTY, any number of independent variables may be defined.

Primary Key Attributes:

FEM ID (FK)

This attribute represents the unique identifier of a FEM. Data Type: String
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK)

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12]. Data Type: Integer

Other Attributes:

FEM MATERIAL TABLE INDEPENDENT VARIABLE DESCRIPTION

Data Type: String

This attribute represents the physical significance of the data contained within the FEM MATERIAL TABLE INDEPENDENT VARIABLE VALUE. For example, the value of this attribute might be 'Temperature'; i.e., 'Temperature' is the physical quantity for which other material properties pointed to by the FEM MATERIAL TABLE INSTANCE data entity are measured with respect to. (See FEM/Issue-21 for a more complete discussion of the FEM MATERIAL TABLE concept.)

Business Rules:

EXPRESS Specification:

```
ENTITY material_table SUBTYPE OF (material_property);
  independent_variable_descriptor       : string;
  material_table_instance_ref    : EXTERNAL LIST [1:#] OF
                     material_table_instance;
END_ENTITY;
```

Entity Name: **FEM MATERIAL TABLE INSTANCE**

Entity Number: **FEM-62**

Data pertaining to an independent variable value and material number is contained within this data entity. It is used in conjunction with a FEM MATERIAL TABLE data entity to complete the specification of tabular material property by associating an independent variable with a pointer to another material property data entity.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

FEM MATERIAL TABLE INDEPENDENT VARIABLE SEQUENCE NUMBER Data Type: Integer
This key attribute is an artifact of the IDEF1X modeling process. Rather than have a real data type for a key attribute, this attribute was artificially created to define a unique key. An ordered list will suffice in EXPRESS.

Other Attributes:

FEM MATERIAL TABLE INDEPENDENT VARIABLE VALUE Data Type: Real
This attribute contains the value of the independent variable used as a reference in the FEM MATERIAL TABLE INSTANCE data entity. (See FEM/Issue-21 for a more complete discussion of the FEM MATERIAL TABLE concept.)

FEM MATERIAL TABLE MATERIAL NUMBER Data Type: Integer
The material table material number points to the material property for the table; i.e.,

it points to the dependent material property data that has been measured with respect to the value of the FEM MATERIAL TABLE INDEPENDENT VARIABLE VALUE. (See FEM/Issue-21 for a more complete discussion of the FEM MATERIAL TABLE concept.)

Business Rules:

EXPRESS Specification:

```
ENTITY material_table_instance;
  material_table_material_ref    : EXTERNAL material_property;
  independent_variable_value     : real;
END_ENTITY;
```

Entity Name: **FEM ISOTROPIC THERMAL EXPANSION**

Entity Number: **FEM-63**

This data entity contains isotropic thermal expansion information.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ISOTROPIC THERMAL EXPANSION COEFFICIENT Data Type: Real

When the temperature of a small portion of an unrestrained isotropic material is changed by an amount, delta T, a change in dilation proportional to delta T is produced without any corresponding change in stress. The ratio of the induced extensional strain to the temperature change, delta T, is the thermal expansion coefficient for the material.

FEM ISOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE Data Type: Real

The thermal expansion reference temperature is the temperature about which all thermal expansion phenomena are related to.

Business Rules:

EXPRESS Specification:

```
ENTITY isotropic_thermal_expansion;
  thermal_expansion_coefficient : real;
  thermal_expansion_reference_temp      : real;
END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC THERMAL EXPANSION**

Entity Number: **FEM-64**

This data entity contains orthotropic thermal expansion information that is common between orthotropic 2D thermal expansion and 3D.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC THERMAL IN-PLANE EXPANSION COEFFICIENTS Data Type: Real
The thermal expansion coefficients, A1 and A2, are the ratios of induced extensional strains in one of the in plane principal material directions due to an applied temperature change, delta T. The coefficient A1 is for the longitudinal direction and A2 applies to expansion in the lateral direction.

FEM ORTHOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE Data Type: Real
The thermal expansion reference temperature is the temperature about which all thermal expansion phenomena are related to.

FEM ORTHOTROPIC THERMAL EXPANSION TYPE Data Type: Enumeration
An enumeration used to identify the type of FEM ORTHOTROPIC THERMAL EXPANSION. The possible types are:

Orthotropic Order Expansion Type	Description
0	2D Orthotropic thermal expansion
1	3D Orthotropic thermal expansion

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_thermal_expansion SUPERTYPE OF (
        orthotropic_2d_thermal_expansion OR
        orthotropic_3d_thermal_expansion);
  in-plane_thermal_expansion_coefficients        : ARRAY [1:2] OF real;
  thermal_expansion_reference_temperature        : real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC 2D THERMAL EXPANSION**

Entity Number: **FEM-65**

This data entity contains anisotropic 2D thermal expansion information.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC 2D THERMAL EXPANSION COEFFICIENTS Data Type: Real

(A1, A2, A12) The ratios of the induced extensional strains to the temperature change, delta T, for an element with 2D anisotropic material are the thermal expansion coefficients for the respective material directions.

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_2d_thermal_expansion SUBTYPE OF (
  anisotropic_thermal_expansion);
  thermal_expansion_coefficients        : ARRAY [1:3] OF real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC 3D THERMAL EXPANSION**

Entity Number: **FEM-66**

This data entity contains anisotropic 3D thermal expansion information.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC 3D THERMAL EXPANSION COEFFICIENTS Data Type: Real
(A1, A2, A3, A4, A5, A6) ???? The ratios of the induced extensional strains to the temperature change, delta T, for an anisotropic material are the thermal expansion coefficients A1, A2 and A3 for the material.

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_3d_thermal_expansion SUBTYPE OF
  (anisotropic_thermal_expansion);
  thermal_expansion_coefficients        : ARRAY [1:6] OF real;
END_ENTITY;
```

Entity Name: **FEM LINE ELEMENT GEOMETRIC PROPERTY**

Entity Number: **FEM-67**

This data entity represents a FEM line elements' (spring and damper) geometric properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY line_geometric_property SUBTYPE OF (geometric_property);
  spring_property_ref    : OPTIONAL EXTERNAL
                   spring_geometric_property;
  damper_property_ref    : OPTIONAL EXTERNAL
                   damper_geometric_property;
WHERE
  NOT((spring_property_ref = NULL) AND (damper_property_ref = NULL));
END_ENTITY;
```

Entity Name: **FEM SPRING PROPERTY**

Entity Number: **FEM-68**

This data entity contains the geometric property values for a FEM spring element.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].

FEM SPRING DEGREE OF FREEDOM SEQUENCE NUMBER/1 Data Type: Integer

FEM SPRING DEGREE OF FREEDOM SEQUENCE NUMBER/2 Data Type: Integer

These two attributes represent a pair of values that are interpreted as nodal degree of freedoms in terms of degree of freedom sequence numbers. They are related to each other by the value of the spring coefficient attribute that is instanced in the FEM SPRING PROPERTY entity. The direction of the degree of freedom sequence number is interpreted through the nodal output coordinate system at the respective damper element ends (1 and 2).

Other Attributes:

FEM SPRING COEFFICIENT Data Type: Real

This attribute contains the value of the spring coefficient. This is the coefficient k in the spring equation Force = k*displacement

FEM SPRING STRESS RECOVERY COEFFICIENT Data Type: Real

The value to be used in the stress recovery of a spring finite element. The spring stress recovery coefficient is k in the equation.

Business Rules:

EXPRESS Specification:

```
ENTITY spring_property;
  dof_sequence_no_1     : integer;
  dof_sequence_no_2     : integer;
  spring_coefficient    : real;
  stress_recovery_coefficient    : real;
END_ENTITY;
```

Entity Name: **FEM DAMPER PROPERTY**

Entity Number: **FEM-69**

This data entity contians the geomtric property valus for a FEM damper element.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

FEM DAMPER DEGREE OF FREEDOM SEQUENCE NUMBER/1 Data Type: Integer

FEM DAMPER DEGREE OF FREEDOM SEQUENCE NUMBER/2 Data Type: Integer

These two attributes represent a pair of values that are interpreted as nodal degree of freedoms in terms of degree of freedom sequence numbers. They are related to each other by the value of the damper coefficient attribute that is instanced in the FEM DAMPER PROPERTY entity. The direction of the degree of freedom sequence number is interpreted through the nodal output coordinate system at the respective damper element ends (1 and 2).

Other Attributes:

FEM DAMPER COEFFICIENT Data Type: Real

The value of the damping coefficient. This is the coefficient k in the damping equation Force = k*velocity

Business Rules:

EXPRESS Specification:

```
ENTITY damper_property;
  dof_sequence_no_1      : integer;
  dof_sequence_no_2      : integer;
  damper_coefficient     : real;
END_ENTITY;
```

Entity Name: **FEM BEAM INTERVAL**

Entity Number: **FEM-70**

This data entity represents an instance of a stiffness interval for a FEM BEAM element. Within this data model, FEM beam elements are permitted to have multiple stiffness intervals that can have step changes in beam geometric properties, such as cross-sectional area, at the ends of these intervals (see FEM/Issue-28). To allow for this type of idealization, the following data values are necessary:

1. The interval lenght fraction at the end of a stiffness interval (referenced as FEM BEAM INTERVAL LENGHT FRACTION/B)
2. The interval's FEM BEAM SECTION are chosen for both the start of the interval and for the end.
3. The FEM MATERIAL PROPERTY chosen for the interval.
   The beam interval is assumed to follow the following:
   (a) The first interval, FEM BEAM INTERVAL SEQUENCE NUMBER of one (1), is assumed to start at a length fraction of zero (0.0) which is located at end one of the beam.
   (b) The interval section is assumed to be constant if the start and end section id are the same otherwise it is assumed to vary linearly over the interval.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

FEM BEAM INTERVAL SEQUENCE NUMBER Data Type: Integer
This number indicates the position for this beam interval in an order list of of beam interval that are needed in order to define the geometric properties of a FEM beam element. FEM BEAM INTERVAL SEQUENCE NUMBER with the value of one (1) is located at the start of the beam (near end 1) and its starting location is beam length fraction 0.0 and ends at the value in FEM BEAM LENGHT FRACTION AT END OF INTERVAL. For all other FEM BEAM INTERVAL SEQUENCE NUMBER, they are at beam lenght fraction as defined in the previous FEM BEAM INTERVAL SEQUENCE NUMBER and end at its value of FEM BEAM LENGHT FRACTION AT END OF INTERVAL.

Other Attributes:

FEM BEAM INTERVAL LENGTH FRACTION /B Data Type: Real
The value of this attribute is the position along a beam finite element at which the end of the interval is defined.

The length fraction is defined as being 0.0 at the first node of the beam finite element and being 1.0 at the end opposite from the first node.

FEM BEAM SECTION NUMBER/A (FK) Data Type: Integer
The identifier of the FEM BEAM SECTION used to defined the start beam section or end A used in the interval defined in the entity FEM BEAM INTERVAL [FEM-70].

See more on this attribute definition in entity FEM BEAM SECTION [Entity FEM-70]

FEM BEAM SECTION NUMBER/B (FK) Data Type: Integer

The identifier of the FEM BEAM SECTION used to defined the ending beam section or end B used in the interval defined in the entity FEM BEAM INTERVAL [FEM-70].

See more on this attribute definition in entity FEM BEAM SECTION [Entity FEM-70].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Business Rules:

EXPRESS Specification:

```
ENTITY beam_interval;
  beam_length_fraction_B         : real;
  section_ref_A  : EXTERNAL beam_section;
  section_ref_B  : EXTERNAL beam_section;
  material_ref   : EXTERNAL material_property;
WHERE
  beam_length_fraction_B BETWEEN 0. AND 1.;
END_ENTITY;
```

Entity Name: **FEM BEAM SECTION**

Entity Number: **FEM-71**

This entities data resprents a beam section to be use in a FEM BEAM INTERVAL entity (FEM-70).

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.

See attribute definition in entity FEM [Entity FEM-1].

FEM BEAM SECTION NUMBER Data Type: Integer

The identifier used to identify a FEM BEAM SECTION from all other within a FEM. It is unique within the current FEM only.

Other Attributes:

FEM BEAM SECTION TYPE Data Type: Enumeration

An enumeration used to identify the type of FEM BEAM SECTION. The possible types are:

Beam Section Type	Description
0	General FEM beam section which means use FEM beam geometric properties.
1	Standard beam section which means use information from standard section to generate beam geometric properties.

Business Rules:

EXPRESS Specification:

```
ENTITY beam_section SUPERTYPE OF (
        beam_property_data OR
        standard_beam_section);
  stress_recovery_coefficients  : OPTIONAL LIST [1:#] OF
                coordinate_pair;
END_ENTITY;
```

Entity Name: **FEM BEAM STANDARD SECTION**

Entity Number: **FEM-72**

This entities data resprents the selection of a standard section for the FEM BEAM SECTION (entity FEM-71). The standard section are defined in a entity STANDARD SECTION. (The STANDARD SECTION entity is outside the scope of the FEM model).

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM BEAM SECTION NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM BEAM SECTION [Entity FEM-71].

Other Attributes:

STANDARD SECTION ID (FK) Data Type: String

The identifier used to identify a STANDARD SECTION that was chosen to be a beam section for this FEM.
See attribute definition in entity STANDARD SECTION [Entity SS-1].

Business Rules:

EXPRESS Specification:

This express entity was merged into the FEM BEAM SECTION entity.

Entity Name: **FEM BEAM PIN DATA**

Entity Number: **FEM-73**

This data entity contains an array of degree of freedoms that the analyst wishes to release (or pin) in a FEM beam element.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM BEAM PIN ARRAY/1 Data Type: Integer
The degree of freedom the analyst wishes to release in the finite element at end 1. This degree of freedom is referenced to the local element coordinate system. The array is defined as follows:

Array Location	Degree of Freedom
1	X translation at end 1 (axial)
2	Y translation at end 1 (y shear)
3	Z translation at end 1 (z shear)
4	X rotation at end 1 (torsion)
5	Y rotation at end 1 (y moment)
6	Z rotation at end 1 (z moment)

FEM BEAM PIN ARRAY/2 Data Type: Integer
The degree of freedom the analyst wishes to release in the finite element at end 2. This degree of freedom is referenced to the local element coordinate system. The array is defined as follows:

Array Location	Degree of Freedom
1	X translation at end 2 (axial)
2	Y translation at end 2 (y shear)
3	Z translation at end 2 (z shear)
4	X rotation at end 2 (torsion)
5	Y rotation at end 2 (y moment)
6	Z rotation at end 2 (z moment)

Business Rules:

EXPRESS Specification:

This express entity was merged into FEM BEAM GEOMETRIC PROPERTY. See entity FEM BEAM GEOMETRIC PROPERTY [Entity FEM-27].

Entity Name: **FEM BEAM OFFSET VECTORS**

Entity Number: **FEM-74**

This data entity contains two vectors that the analyst wish to use in offsetting the FEM beam's shear center at end one (1) and end two (2) respectively.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM BEAM OFFSET VECTOR END 1 Data Type: Real
Three coordinates, x, y, and z, referenced to the output coordinate system of the node at end 1 of the beam, are used to define the amount by which the beam's neutral axis is offset from the node at end 1 of the beam.

FEM BEAM OFFSET VECTOR END 2 Data Type: Real
Three coordinates, x, y, and z, referenced to the output coordinate system of the node at end 2 of the beam, are used to define the amount by which the beam's neutral axis is offset from the node at end 2 of the beam.

Business Rules:

EXPRESS Specification:

This express entity was merged into FEM BEAM GEOMETRIC PROPERTY. See entity FEM BEAM GEOMETRIC PROPERTY [Entity FEM-27].

Entity Name: **FEM BEAM WARPING DATA**

Entity Number: **FEM-75**

This data entity contains two warping coefficient to be applied to the FEM beam element at end one (1) and end two (2) respectively.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM GEOMETRIC PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM GEOMETRIC PROPERTY [Entity FEM-11].

Other Attributes:

FEM BEAM WARPING COEFFICIENT END 1 Data Type: Real
The FEM beam element warping coefficient at end one (1) of the beam.

FEM BEAM WARPING COEFFICIENT END 2 Data Type: Real
The FEM beam element warping coefficient at end two (2) of the beam.

Business Rules:

EXPRESS Specification:

This express entity was merged into FEM BEAM GEOMETRIC PROPERTY. See entity FEM BEAM GEOMETRIC PROPERTY [Entity FEM-27].

Entity Name: **FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY**

Entity Number: **FEM-76**

This data entity represents the FEM orthotropic structural material properties that can be either 2D or 3D orthotropic structural material properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC STRUCTURAL IN-PLANE POISSON'S RATIO Data Type: Real
The in-plane Poisson's ratio, v12, for an orthotropic material represents the ratio of lateral contraction to longitudinal extension in a simple in-plane longitudinal tension test.

FEM ORTHOTROPIC STRUCTURAL IN-PLANE SHEAR MODULUS Data Type: Real
The in-plane shear modulus, G12, of an orthotropic material represents the ratio of shearing stress to shearing strain for in-plane shearing deformations.

FEM ORTHOTROPIC STRUCTURAL IN-PLANE YOUNG'S MODULI Data Type: Real
Young's modulus in the longitudinal direction for an orthotropic elastic material is the ratio of longitudinal stress to longitudinal strain observed in a simple longitudinal tension test with traction free transverse and lateral surfaces. Young's modulus for the longitudinal direction is denoted by E11.
Young's modulus in the lateral direction for an orthotropic elastic material is the ratio of stress to strain observed in a simple lateral tension test with traction free longitudinal and transverse surfaces. Young's modulus for the lateral direction is denoted by E22.

FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE Data Type: Enumeration
An enumeration used to identify the type of FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY. The possible types are:

Orthotropic Structural Material Property Type	**Description**
0	2D Orthotropic structural material property
1	3D Orthotropic structural material property

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_structural_property SUPERTYPE OF (
       orthotropic_structural_2d_property OR
```

```
         orthotropic_structural_3d_property);
    in-plane_youngs_moduli        : ARRAY [1:2] OF real;
    in-plane_poissons_ratio       : real;
    in-plane_shear_modulus        : real;
  END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY**

Entity Number: **FEM-77**

This data entity represents the FEM orthotropic thermal material properties that can be either 2D or 3D orthotropic thermal material properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC THERMAL CONVECTIVE FILM COEFFICIENT Data Type: Real
The convective film coefficient is the constant of proportionality that relates the magnitude of the heat flux normal to an external material boundary to the temperature difference occurring across the boundary.

FEM ORTHOTROPIC THERMAL HEAT CAPACITY Data Type: Real
The heat capacity of a material is the total quantity of heat required to produce a one unit of change of temperature for a unit mass of the material at constant pressure. The heat capacity is a scalar quantity independent of direction for all material types, whether isotropic, orthotropic, or anisotropic.

FEM ORTHOTROPIC THERMAL IN-PLANE THERMAL CONDUCTIVITIES Data -Type: Real
K1 and K2 are the in-plane thermal conductivities.

FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY TYPE Data Type: Enumeration
An enumeration used to identify the type of FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY. The possible types are:

Orthotropic Thermal Material Property Type	Description
0	2D Orthotropic thermal material property
1	3D Orthotropic thermal material property

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_thermal_property SUPERTYPE OF (
        orthotropic_thermal_2d_property OR
        orthotropic_thermal_3d_property);
  in-plane_thermal_conductivities       : ARRAY [1:2] OF real;
  specific_heat_capacity        : real;
  convective_film_coefficient   : real;
END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC 2D THERMAL EXPANSION**

Entity Number: **FEM-78**

This data entity contains the values for orthotropic 2D thermal expansion. Although currently there are no attributes in this entity, it was added for clarity reasons.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

None

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_2d_thermal_expansion SUBTYPE OF
  (orthotropic_thermal_expansion);
END_ENTITY;
```

Entity Name: **FEM ORTHOTROPIC 3D THERMAL EXPANSION**

Entity Number: **FEM-79**

This data entity contains the additional values needed for orthotropic 3D thermal expansion above those for 2D.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ORTHOTROPIC THERMAL OUT-PLANE EXPANSION COEFFICIENTS Data Type: Real
The thermal expansion coefficients, A3, is the ratios of induced extensional strains in the out-plane principal material directions due to an applied temperature change, delta T. The coefficient A3 is for the transver direction.

Business Rules:

EXPRESS Specification:

```
ENTITY orthotropic_3d_thermal_expansion SUBTYPE OF
       (orthotropic_thermal_expansion);
  out-of-plane_thermal_expansion_coefficient    : real;
END_ENTITY;
```

Entity Name: FEM ANISOTROPIC THERMAL MATERIAL PROPERTY

Entity Number: FEM-80

This data entity represents the FEM anisotropic thermal material properties that can be either 2D or 3D anisotropic thermal material properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String

This attribute represents the unique identifier of a FEM. See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer

See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC THERMAL HEAT CAPACITY Data Type: Real

The heat capacity of a material is the total quantity of heat required to produce a one unit of change of temperature for a unit mass of the material at constant pressure. The heat capacity is a scalar quantity independent of direction for all material types, whether isotropic, orthotropic, or anisotropic.

FEM ANISOTROPIC THERMAL MATERIAL PROPERTY TYPE Data Type: Enumeration

An enumeration used to identify the type of FEM ANISOTROPIC THERMAL MATERIAL PROPERTY. The possible types are:

Anisotropic Thermal Material Property Type	Description
0	2D Anisotropic thermal material property
1	3D Anisotropic thermal material property

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_thermal_property SUPERTYPE OF (
        anisotropic_2d_thermal_property OR
        anisotropic_thermal_3d_property);
  specific_heat_capacity          : real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC THERMAL EXPANSION**

Entity Number: **FEM-81**

This data entity represents the FEM anisotropic thermal expansion material properties that can be either 2D or 3D anisotropic thermal expansion material properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC THERMAL EXPANSION REFERENCE TEMPERATURE Data Type: Real
The thermal expansion reference temperature is the temperature about which all thermal expansion phenomena are related to.

FEM ANISOTROPIC THERMAL EXPANSION TYPE Data Type: Enumeration
An enumeration used to identify the type of FEM ANISOTROPIC THERMAL EXPANSION. The possible types are:

Anisotropic Thermal Expansion Type	Description
0	2D Anisotropic thermal expansion
1	3D Anisotropic thermal expansion

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_thermal_expansion SUPERTYPE OF (
        anisotropic_2d_thermal_expansion OR
        anisotropic_3d_thermal_expansion);
  thermal_expansion_reference_temp      : real;
END_ENTITY;
```

Entity Name: **FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY**

Entity Number: **FEM-82**

This data entity represents the FEM anisotropic structural material properties that can be either 2D or 3D anisotropic structural material properties.

Primary Key Attributes:

FEM ID (FK) Data Type: String
This attribute represents the unique identifier of a FEM.
See attribute definition in entity FEM [Entity FEM-1].

FEM MATERIAL PROPERTY NUMBER (FK) Data Type: Integer
See attribute definition in entity FEM MATERIAL PROPERTY [Entity FEM-12].

Other Attributes:

FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY TYPE Data Type: Enumeration
An enumeration used to identify the type of FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY. The possible types are:

Anisotropic Structural Material Property Type	Description
0	2D Anisotropic structural material property
1	3D Anisotropic structural material property

Business Rules:

EXPRESS Specification:

```
ENTITY anisotropic_structural_property SUPERTYPE OF (
        anisotropic_structural_2d_property OR
        anisotropic_structural_3d_property);
END_ENTITY;
```

Entity Name: **PRODUCT ITEM VERSION**

Entity Number: **PSCM-4**

A PRODUCT ITEM VERSION is an entity that the enterprise keeps track of; e.g., a part, a drawing, an assembly, etc...

Please see section ?? of this document for more details.

Primary Key Attributes:

PRODUCT ITEM ID — Data Type: Integer

This attribute contains the unique identifier for a PRODUCT ITEM. See attribute definition in entity PRODCUT ITEM VERSION [Entity PSCM-4] for complete details.

Other Attributes:

Unknown

Business Rules:

EXPRESS Specification:

See Entity PSCM-4 in section ?? of this document.

Entity Name: **STANDARD SECTION**

Entity Number: **SS-1**

A STANDARD SECTION is a reference to data outside the scope of a FEM that contains standard beam cross-section information.

Primary Key Attributes:

STANDARD SECTION ID — Data Type: Integer

The identifier used to identify a standard section to be used as a FEM BEAM SECTION.

Other Attributes:

Unknown

Business Rules:

EXPRESS Specification:

```
ENTITY standard_beam_section SUBTYPE OF (beam_section);
```

```
    property_data : external_reference;
  END_ENTITY;
```

Entity Name: UNITS

Entity Number: UNIT-1

The UNITS data entity defines the base (or fundamental) units of measure that are to be use and for deriving the other units of measure. For example, A finite element model would have one of these where all the data in the model would be expressed in the nine fundamental units of measure defined by the units entity.

For more information on this entity and other unit entities, please see section ?? of this document.

There are two set of standard fundamental units of measure. They are as follows with the values of the UNITS entity attributes:

Standard set of fundamental units of measure - SI

Attributes	Values
LENGTH UNIT SCALE FACTOR	1.0
BASE LENGTH UNIT	Meter
MASS UNIT SCALE FACTOR	1.0
BASE MASS UNIT	Kilogram
TIME UNIT SCALE FACTOR	1.0
BASE TIME UNIT	Second
ELECTRIC CURRENT UNIT SCALE FACTOR	1.0
BASE ELECTRIC CURRENT UNIT	Ampere
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR	1.0
BASE THERMODYNAMIC TEMPERATURE UNIT	Kelvin
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR	1.0
BASE AMOUNT OF SUBSTANCE UNIT	Mole
LUMINOUS INTENSITY UNIT SCALE FACTOR	1.0
BASE LUMINOUS INTENSITY UNIT	Candela
PLANE ANGLE UNIT SCALE FACTOR	1.0
BASE PLANE ANGLE UNIT	Radian
SOLID ANGLE UNIT SCALE FACTOR	1.0
BASE SOLID ANGLE UNIT	Steradian

Standard set of fundamental units of measure - British

Attributes	Values
LENGTH UNIT SCALE FACTOR	1.0
BASE LENGTH UNIT	Foot
MASS UNIT SCALE FACTOR	1.0
BASE MASS UNIT	Slug
TIME UNIT SCALE FACTOR	1.0
BASE TIME UNIT	Second
ELECTRIC CURRENT UNIT SCALE FACTOR	1.0
BASE ELECTRIC CURRENT UNIT	Ampere
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR	1.0
BASE THERMODYNAMIC TEMPERATURE UNIT	Rakine
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR	1.0
BASE AMOUNT OF SUBSTANCE UNIT	Mole
LUMINOUS INTENSITY UNIT SCALE FACTOR	1.0
BASE LUMINOUS INTENSITY UNIT	Candela
PLANE ANGLE UNIT SCALE FACTOR	1.0
BASE PLANE ANGLE UNIT	Degree
SOLID ANGLE UNIT SCALE FACTOR	1.0
BASE SOLID ANGLE UNIT	Steradian

Primary Key Attributes:

UNITS ID Data Type: String

This attribute contains the unique identifier of the fundamental UNITS.

Other Attributes:

LENGTH UNIT SCALE FACTOR (FK) Data Type: Real

See attribute definition in entity SCALED LENGTH UNIT [Entity UNIT-2]

BASE LENGTH UNIT (FK) Data Type: Enumeration

See attribute definition in entity SCALED LENGTH UNIT [Entity UNIT-2]

MASS UNIT SCALE FACTOR (FK) Data Type: Real

See attribute definition in entity SCALED MASS UNIT [Entity UNIT-3]

BASE MASS UNIT (FK) Data Type: Enumeration

See attribute definition in entity SCALED MASS UNIT [Entity UNIT-3]

TIME UNIT SCALE FACTOR (FK) Data Type: Real

See attribute definition in entity SCALED TIME UNIT [Entity UNIT-4]

BASE TIME UNIT (FK) Data Type: Enumeration

See attribute definition in entity SCALED TIME UNIT [Entity UNIT-4]

ELECTRIC CURRENT UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED ELECTRIC CURRENT UNIT [Entity UNIT-5]

BASE ELECTRIC CURRENT UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED ELECTRIC CURRENT UNIT [Entity UNIT-5]

THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED THERMODYNAMIC TEMPERATURE UNIT [Entity UNIT-6]

BASE THERMODYNAMIC TEMPERATURE UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED THERMODYNAMIC TEMPERATURE UNIT [Entity UNIT-6]

AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED AMOUNT OF SUBSTANCE UNIT [Entity UNIT-7]

BASE AMOUNT OF SUBSTANCE UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED AMOUNT OF SUBSTANCE UNIT [Entity UNIT-7]

LUMINOUS INTENSITY UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED LUMINOUS INTENSITY UNIT [Entity UNIT-8]

BASE LUMINOUS INTENSITY UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED LUMINOUS INTENSITY UNIT [Entity UNIT-8]

PLANE ANGLE UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED PLANE ANGLE UNIT [Entity UNIT-9]

BASE PLANE ANGLE UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED PLANE ANGLE UNIT [Entity UNIT-9]

SOLID ANGLE UNIT SCALE FACTOR (FK) Data Type: Real
See attribute definition in entity SCALED SOLID ANGLE UNIT [Entity UNIT-10]

BASE SOLID ANGLE UNIT (FK) Data Type: Enumeration
See attribute definition in entity SCALED SOLID ANGLE UNIT [Entity UNIT-10]

Business Rules:

EXPRESS Specification:

```
ENTITY units;
  scaled_length_unit      : INTERNAL scaled_length_unit;
  scaled_mass_unit        : INTERNAL scaled_mass_unit;
```

```
    scaled_time_unit          : INTERNAL scaled_time_unit;
    scaled_electric_current_unit    : INTERNAL
                      scaled_electric_current_unit;
    scaled_thermodynamic_temperature_unit  : INTERNAL
          scaled_thermodynamic_temperature_unit;
    scaled_amount_of_substance_unit        : INTERNAL
                      scaled_amount_of_substance_unit;
    scaled_luminous_intensity_unit         : INTERNAL
                      scaled_luminous_intensity_unit;
    scaled_plane_angle_unit     : INTERNAL scaled_plane_angle_unit;
    scaled_solid_angle_unit     : INTERNAL scaled_solid_angle_unit;
  END_ENTITY;
```

9.4 FEM IDEF0 ACTIVITY MODEL

The purpose of the IDEF0 activity models is to show the FEM process in order to help in the scoping and creation of the reference data models. They are not intended to be detailed, but they do give a general overview of the FEM process.

9.5 FEM Relationship to the Planning Model

The STEP/PDES planning model is shown in Figure 143 The shaded-in entity is the entity that the FEM data model partially expands.

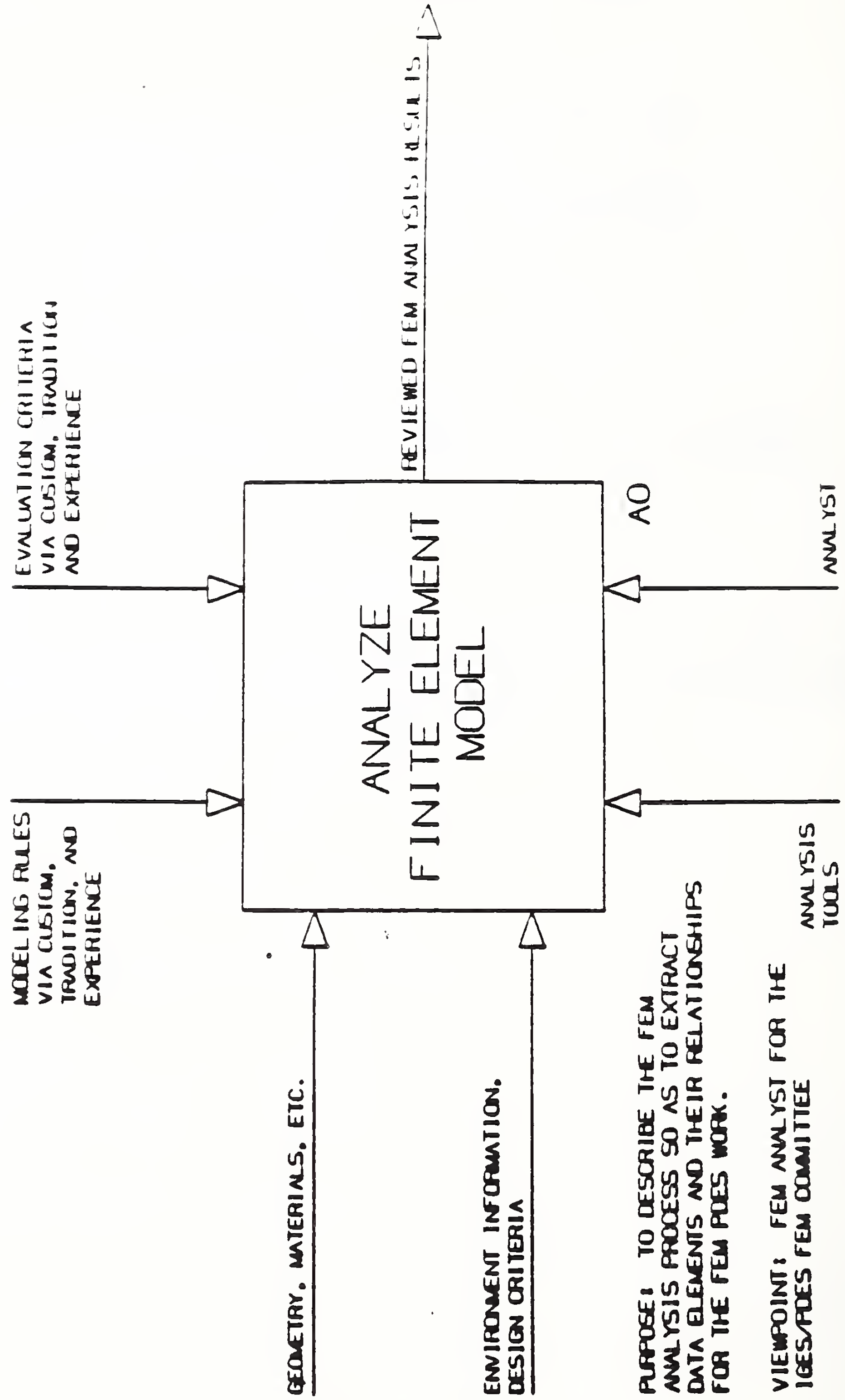

EVALUATION CRITERIA VIA CUSTOM, TRADITION AND EXPERIENCE
REVIEWED FEM ANALYSIS RESULTS
MODELING RULES VIA CUSTOM, TRADITION, AND EXPERIENCE
ANALYZE FINITE ELEMENT MODEL
A0
ANALYST
ANALYSIS TOOLS
GEOMETRY, MATERIALS, ETC.
ENVIRONMENT INFORMATION, DESIGN CRITERIA
PURPOSE: TO DESCRIBE THE FEM ANALYSIS PROCESS SO AS TO EXTRACT DATA ELEMENTS AND THEIR RELATIONSHIPS FOR THE FEM PDES WORK.
VIEWPOINT: FEM ANALYST FOR THE IGES/PDES FEM COMMITTEE

Figure D-141: Context of Analyse FEM

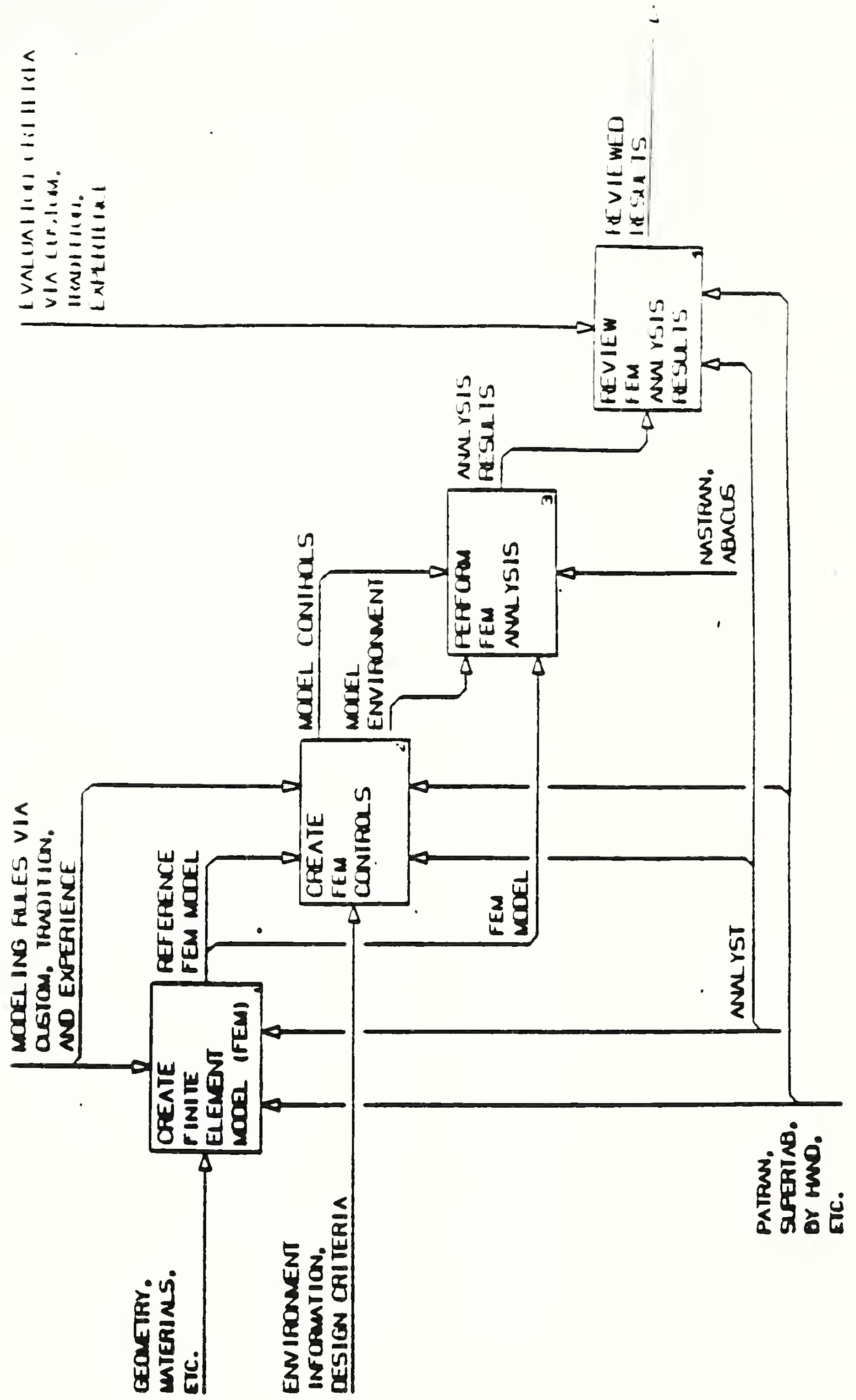


Figure D-142: Analyze Finite Element Model

Figure D–143: FEM Planning Model

SECTION 9: FEM INFORMATION MODEL

9.6 Issue Log

The following list is a table of contents of issues for the FEM data model. Contained in the list are:

1. An issues identify;
2. A single character representing the current status of the issue. An R indicates that the issue is resolved. An U indicates that the issue remains unresolved. An X indicates that the issue is obsolete. An O indicates that the issue remains open and has a temporary solution defined for it. But the work remains or consensus has not been reached yet.
3. A short description is given as the title of the issue

Issue	Status	Description
FEM-1	O	Interaction with curves and surfaces geometry
FEM-2	U	Missing degree of freedom concept
FEM-3	U	Missing special nodes concept
FEM-4	R	Coordinate systems
FEM-5	U	Substructures concept
FEM-6	R	Units
FEM-7	R	Multiple analysis case concept
FEM-8	R	Color group
FEM-9	R	Combined geometric and material properties
FEM-10	R	Stress recovery location data
FEM-11	R	Gauss point integration switch
FEM-12	R	Spring stress coefficients
FEM-13	X	Missing material reference in beam properties
FEM-14	R	Diagonal mass matrix data
FEM-15	R	Node sequence number propagation
FEM-16	U	Element characteristics
FEM-17	R	Composite property data location
FEM-18	R	Beam cross-section definitions
FEM-19	U	FEM, FEA, and FEA CONTROL relationships
FEM-20	R	IDEF1X failure to model ordered lists
FEM-21	R	Tabular material data
FEM-22	R	Material properties for Dynamics Problems
FEM-23	R	Material property cardinality logic
FEM-24	R	Thermal expansion property separation
FEM-25	R	Real Keys

Issue	Status	Description
FEM-26	R	Springs, Dampers, and Beams
FEM-27	R	Membrane Geometric Property
FEM-28	R	FEM Beam Model
FEM-29	R	Preventing fatal recursive Material Id reference
FEM-30	O	FEM Material Allowables
FEM-31	O	Material Property Coordinate System
FEM-32	O	Point Damping Coefficient
FEM-33	O	Time stamp Relationship Between Product Item and FEM
FEM-34	O	Mesh Generation entities
FEM-35	O	Part Geometry And FEM Geometry Are Not Identical
FEM-36	O	FEM Material and Geometric Property Libraries
FEM-37	O	FEM Geometric Property Uniqueness
FEM-38	O	Text for FEM, Material, Control, Groups and Environment
FEM-39	O	Node Ordering for FEM Elements

ISSUE:	FEM–1 Interaction with Curves and Surfaces Geometry.
INITIATION DATE:	04/16/86 and 06/23/87
INITIATOR:	FEM Committee and ISO Subgroup 2
STATUS:	Unresolved
DESCRIPTION:	How does the FEM data model interface with the Curves and Surfaces and/or the Solids data models?
RESOLUTION:	The following candidate resolution to this issue is proposed based upon a joint meeting that was held with the Curves and Surfaces (CS) subgroup during the Summer ISO meeting in 1987. A FEM ELEMENT may be associated with a CS SHELL, FACE, EDGE, and/or VERTEX. Similarly, a FEM NODE may also be associated with anyone of these CS data entities. Figure 144 and 145 depict IDEF1X diagrams of the above interaction with Curves and Surfaces geometry.

ISSUE:	FEM-2 Missing Degree of Freedom Concept.
INITIATION DATE:	07/15/86 and 11/06/86
INITIATOR:	FEM Committee and ISO Subgroup 2
STATUS:	Unresolved
DESCRIPTION:	The FEM data model does not address the concept of nodal degrees of freedom. Is a degree of freedom sequence number attribute in the CONSTRAINT data entity sufficient to implement the concept? How to degree of freedom numbers get associated with what they actually imply? Are business rules sufficient to cover this?

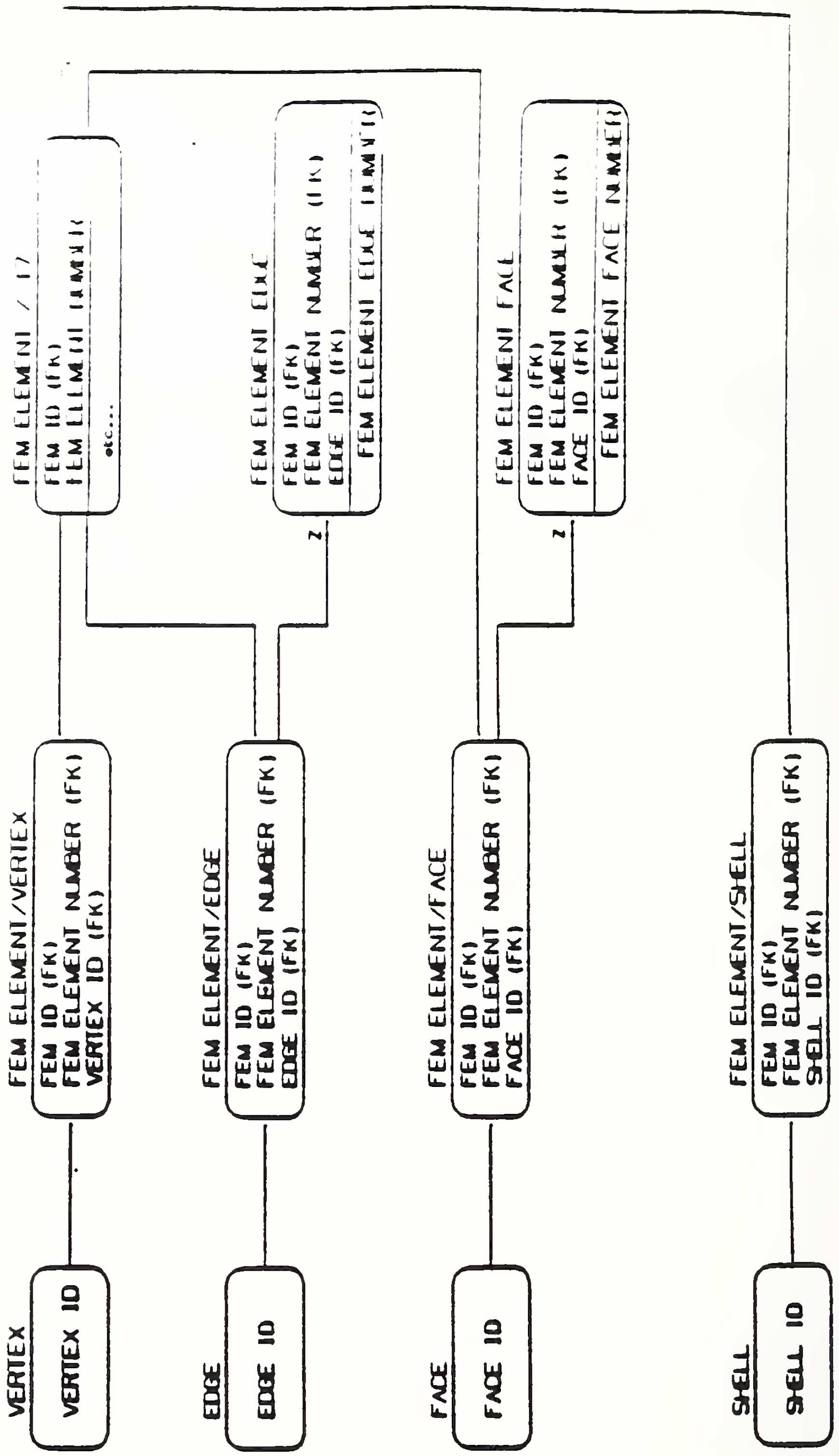


Figure D-144: FEM Element Integration

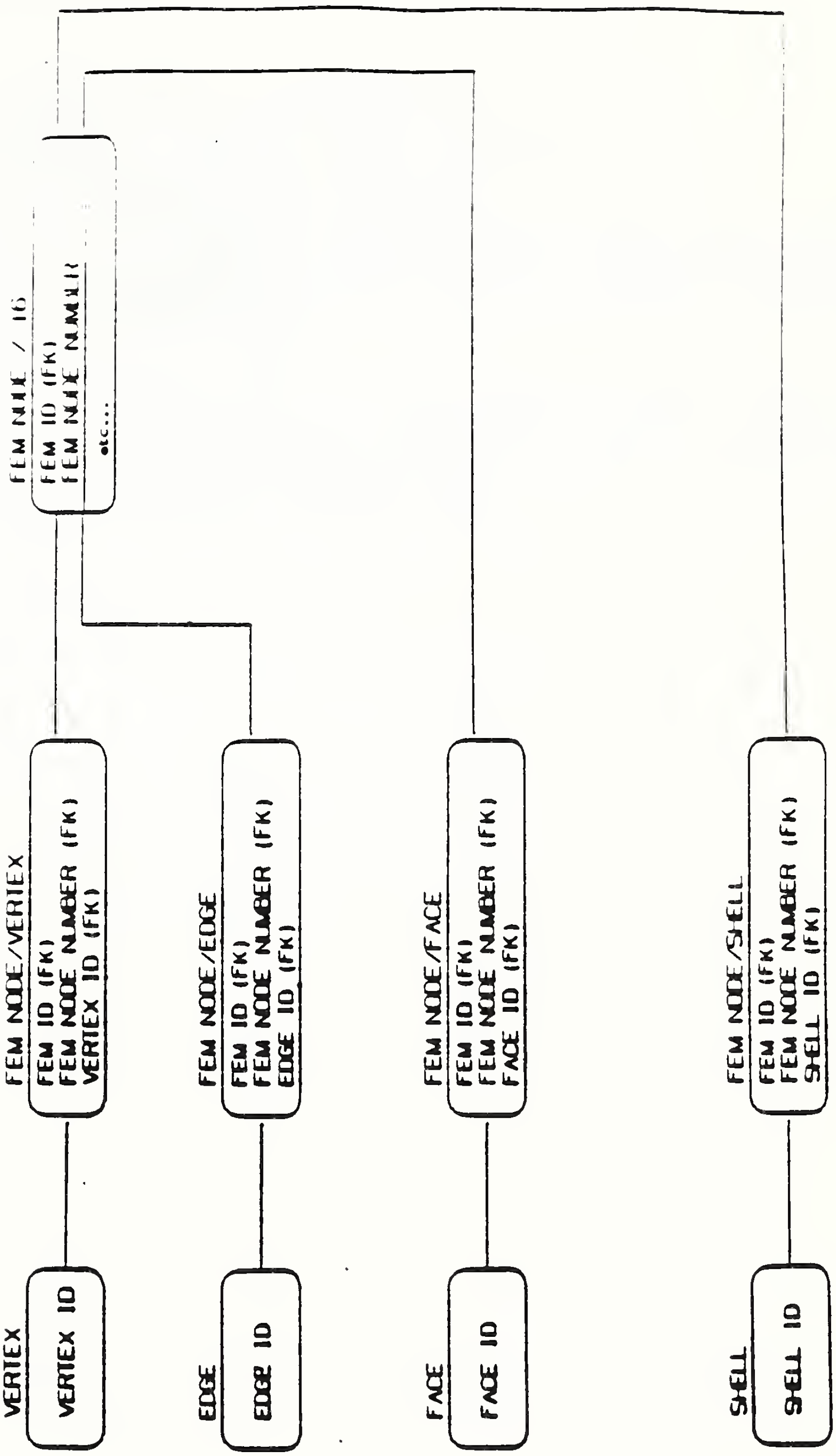


Figure D-145: FEM Node Integration

ISSUE: FEM-3 Missing Special Nodes Concept.
INITIATION DATE: 07/15/86
INITIATOR: FEM Committee
STATUS: Unresolved
DESCRIPTION: The FEM data model does not address the concept of indicating that a node is special; i.e., there is no way to indicate a MSC/NASTRAN STATIC node or a thermal node.

ISSUE: FEM-4 Coordinates Systems.

INITIATION DATE: 07/15/86

INITIATOR: FEM Committee

STATUS: Resolved

DESCRIPTION: Is it legitimate to use COORDINATE SYSTEM IDENTIFICATION NUMBERs as attributes to the FEM ELEMENT and FEM NODE data entity, instead of requiring separate definitions for these special coordinate systems?

RESOLUTION: This point was raised with members of the logical layer and it was suggested that it was legitimate to use COORDINATE SYSTEM IDENTIFICATION NUMBERS as attributes.

DECISION DATE: 07/16/86 by FEM Committee

ISSUE: FEM-5 Substructures Concept

INITIATION DATE: 07/16/86 and 11/02/86

INITIATOR: FEM Committee and ISO Subgroup 2

STATUS: Unresolved - currently out of scope

DESCRIPTION: Is grouping sufficient for passing substructure type data? How does the FEM data model pass MSC/NASTRAN Superelement information? How does it pass PAFEC PAFBLOCK information? Where could information, such as mirroring or replicating of substructures, be represented in the FEM data model?

It would be useful to introduce a new entity into the reference model called a FEM OBJECT (see the CAD*I reference [CAD*I]) which forms a structure for an analysis to act on. It would be useful to have an this entity in the reference model which can be used as a fundamental entity for analysis. The FEM OBJECT would be initially be created from a collection of elements but then could be copied, assembled, and condensed before undergoing analysis. This entity is in addition to the entity FEM GROUP which is useful for the purpose of assigning certain attributes to a collection of entities. Many of the facilities which the FEM OBJECT makes possible are outside of th scope of the present model but including it at this stage will ma future extensions easier.

ISSUE: FEM-6 Units.

INITIATION DATE: 07/16/86

INITIATOR: FEM Committee

STATUS: Resolved

DESCRIPTION: How does the concept of units of measurement get introduced into the STEP data model and consequently how does it interface with the FEM data model? Can a UNITS data entity cover all fundamental units of measure? (Is electrical current a fundamental unit of measure?) Since units of angular measurement are not fundamental units of measure, do they need to be included within the UNITS data entity? Standard practice permits the specification of angular data in either degrees or radians. Angular position data is required for the specification of coordinate positions in either cylindrical or spherical coordinate systems. Therefore, the type of angular data must be identified. The logical place for it to be located is in a UNITS entity.

RESOLUTION: The solution chosen was to create a UNITS data entity. There are seven base units of measurement. These are length, mass, time, electric current, thermodynamic temperature, amount of substance, and luminous intensity. Also, there are two supplemental units. These are plane angles and solid angles. The UNITS data entity has one attribute for each base and supplemental unit of measurement. All floating-point values that have units must be consistently defined within the scope of this data entity.

DECISION DATE: 04/01/87 by ISO Subgroup 2

ISSUE:	FEM-7 Multiple Analysis Case Concept.
INITIATION DATE:	04/10/86
INITIATOR:	FEM Committee
STATUS:	Resolved 07/16/86
DESCRIPTION:	What should the committee's opinion be about how to reference multiple analysis cases from within the same data model, especially if these data are created from separate and distinct computer executions? The FEM data model is currently defined in terms of results as they relate to a single computer execution.

ISSUE OPTIONS & EVIDENCE:

Pro: FEA CONTROL data, such as which load cases are combined together to produce a finite element analysis, is really data.

Con: The inclusion of FEA CONTROL data is process information that is not intended to be modeled via the IDEF1X modeling methodology.

RESOLUTION:	The con argument has not had any champions attend any of the recent meetings. Therefore, this issue has been resolved by default in favor of the pro argument. The FEA CONTROL data entity will have t capability to reference multiple load and constraint cases.
DECISION DATE:	07/16/86 by FEM Committee
NOTE:	Not Diagrammed Yet.

ISSUE:	FEM-8 Colour Group.
INITIATION DATE:	10/20/86
INITIATOR:	FEM Committee
STATUS:	Resolved
DESCRIPTION:	Do FEM COLOR GROUP and FEM NAME GROUP represent the same data concept? If so, should these entities be collapsed into a single entity?
RESOLUTION:	It was decided that the FEM COLOR GROUP and the FEM NAME GROUP represented the same data concept. These two entities were collapsed into the new entity FEM GROUP NAME. The name of this group may be used to transfer color or any other arbitrary naming information.
DECISION DATE:	10/20/86 by FEM Committee

ISSUE:	FEM-9 Combined Geometric and Material Properties.
INITIATION DATE:	10/22/86
INITIATOR:	FEM Committee
STATUS:	Resolved
DESCRIPTION:	How do we reference geometric properties that are combined with material properties such as MSC/NASTRAN's shell elements?
RESOLUTION:	Material properties should be an independent data entity not tied to either the FEM NODE or FEM ELEMENT data entities. The portion of the material properties included within the IDEF1X FEM data model is, however, dependent on the FEM. There will be an external reference to some, as yet undefined, material property entity. Model has been updated to reflect this decision.
DECISION DATE:	10/22/86 by FEM Committee

ISSUE:	FEM-10 Stress Recovery Location Data.
INITIATION DATE:	10/22/86
INITIATOR:	FEM Committee
STATUS:	Resolved
DESCRIPTION:	Where does stress recovery location data fit into the model?
RESOLUTION:	An additional entity (Entity 28 - FEM SHELL STRESS RECOVERY COEFFICIENT) was added to the FEM reference model, see Figure 114. A FEM SHELL GEOMETRIC PROPERTY references the FEM SHELL STRESS RECOVERY COEFFICIENTs. FEM BEAM STRESS RECOVERY COEFFICIENTs were also added.
DECISION DATE:	01/20/87 by FEM Committee

ISSUE: FEM-11 Gauss Point Integration Switch.
INITIATION DATE: 10/22/86
INITIATOR: FEM Committee
STATUS: Resolved
DESCRIPTION: How can the concept of a gauss point integration switch be incorporated into the data model? Is there other data contained on an MSC/NASTRAN PSOLID card that should be incorporated into the FEM data model?

RESOLUTION: An additional entity (Entity 55 - FEM INTEGRATION ORDER) was added to the FEM reference model, see Figure 115. A FEM GEOMETRIC PROPERTY references the FEM INTEGRATION ORDER.
DECISION DATE: 01/20/87 by FEM Committee

ISSUE:	FEM-12 Spring Stress Recovery Coefficients,
INITIATION DATE:	10/22/86
INITIATOR:	FEM Committee
STATUS:	Resolved
DESCRIPTION:	The concept of stress recovery coefficients for spring finite elements is missing from the FEM data model?
RESOLUTION:	An additional entity (Entity FEM-34 - FEM SPRING STRESS RECOVERY COEFFICIENTS) was added to the FEM reference model, see Figure 112. A FEM SPRING PROPERTY references the FEM SPRING STRESS RECOVERY COEFFICIENTS data entity.
DECISION DATE:	04/01/87 by ISO Subgroup 2

ISSUE:	FEM-13 Missing Material Reference in Beam Properties.
INITIATION DATE:	11/04/86
INITIATOR:	ISO Subgroup 2
STATUS:	Resolved
DESCRIPTION:	Material references are missing in the BEAM PROPERTIES data entity.
RESOLUTION:	An additional attribute was added to the FEM BEAM PROPERTY data entity (Entity 33), see Figure 112. The attribute that was added is the FEM BEAM MATERIAL NUMBER, a foreign key.
DECISION DATE:	01/20/87 by FEM Committee
NOTE:	Resolution of Issue FEM-28 made the above issue obsolete.

ISSUE:	FEM-14 Diagonal Mass Matrix Data.
INITIATION DATE:	11/04/86
INITIATOR:	ISO Subgroup 2
RELATED ISSUES:	FEM-18
STATUS:	Resolved
DESCRIPTION:	The current IDEF1X data model does not permit the direct specification of a diagonal mass matrix. It can be indirectly be specified by using the full 6x6 mass matrix. Why are both offset mass and 6x6 mass matrix data being specified? Isn't this a duplication of application data analogous to the arguments in favor of eliminating beam cross-sectional geometry information? We need to be internally consistent.
RESOLUTION:	This issue was resolved in favor of being internally consistent. Two data entities were deleted from the FEM reference model that dealt with variant forms of finite elemental mass matrices. The model currently only permits the specification of a 6x6 mass matrix at a node via the FEM POINT GEOMETRIC PROPERTY data entity (Entity 26), see Figure 110.
DECISION DATE:	03/30/87 by ISO Subgroup 2

ISSUE:	FEM-15 Node Sequence Number Propagation.
INITIATION DATE:	11/05/86
INITIATOR:	ISO Subgroup 2
STATUS:	Resolved
DESCRIPTION:	After the 1986 Huntsville IGES meeting, several new and expanded IDEF1X model diagrams were developed. Among the new ones developed was the FEM Geometric Properties model. It was the consensus of the ISO subgroup that a problem exists with 1986 Huntsville model. The problem occurs in modeling the relationship that some shell finite elements have which permit varying elemental geometric data, such as elemental thickness. These data are usually associated with an element's nodes and it was originally modeled by relating the FEM NODE SEQUENCE NUMBER to numerous of the FEM Geometric Property data entities on our IDEF1X diagram. Apart from the myriad of connecting lines that this decision produced, no one in the subgroup could convince themselves that the model was correct as represented. Many solutions were proposed which after exploration did not solve the problem. The only thing that could be said of the alternative solutions was that they only reinforced our conviction that the Huntsville model was wrong with regards to this item.
RESOLUTION:	Rename the FEM NODE GEOMETRIC PROPERTY data entity into the FEM ELEMENT/NODE GEOMETRIC PROPERTY entity and connect this new data entity with the FEM CONNECTIVITY data entity. Each FEM CONNECTIVITY refers to zero or one FEM ELEMENT/NODE GEOMETRIC PROPERTIES. Once this is done all data paths between FEM CONNECTIVITY and the various FEM properties can be removed, as well as the propagation of the FEM ELEMENT NUMBER and FEM NODE SEQUENCE NUMBER foreign keys into these data entities. Further, the varying shell element property type can be completely removed from the IDEF1X diagram since it is now handled differently. The current model for handling the FEM properties that vary at a finite element's nodes is shown in Figure 108.
DECISION DATE:	01/20/87 by FEM Committee

ISSUE: FEM-16 Element Characteristics.
INITIATION DATE: 04/14/85
INITIATOR: FEM Committee
STATUS: Unresolved
DESCRIPTION:

RESOLUTION: (Writeup about Fong's paper)

ISSUE: FEM-17 Composite Property Data Location.
INITIATION DATE: 11/04/86
INITIATOR: ISO Subgroup 2
STATUS: Resolved
DESCRIPTION: The ISO working raised a question about the location of the FEM COMPOSITE MATERIAL PROPERTY in the FEM reference model. Should it be located as a sub-type of FEM MATERIAL PROPERTY (Entity 12) or should it be renamed and be located as a sub-type of FEM GEOMETRIC PROPERTY (Entity 11)? the FE

ISSUE OPTIONS & EVIDENCE:

Pro: When working with composite materials, it is common practice to refer to them as materials and not as geometric properties of different materials, therefore the location of the FEM COMPOSITE MATERIAL PROPERTY should not change.

Con: Ideas like volume fraction, volume orientation, ply sequence number, and ply thickness seam to be more geometrical in nature, therefore the FEM COMPOSITE MATERIAL PROPERTY should be renamed and relocated to the geometric properties portion of the FEM reference model.

RESOLUTION: The pro argument was considered to be the answer. When dealing with composite materials, they are referred to as material properties and not as geometric properties, see Figure 117.
DECISION DATE: 1/15/87 by FEM Committee

ISSUE: FEM-18 Beam Cross-section Definitions.

INITIATION DATE: 10/20/86

INITIATOR: FEM Committee

STATUS: Resolved

DESCRIPTION: Two candidate models were presented to the working group that described the concept of having FEM BEAM PROPERTY sub-types based upon beam cross-section. The question is should beam cross-section geometry be defined by the FEM reference model?

RESOLUTION: It was decided that beam cross-section geometry should not be defined by the FEM reference model. The fundamental data required by a FEA is the beam second moments of area and the cross-sectional area, not the geometry that produced the fundamental data. It is believed that all analysis codes could accept fundamental beam data of this type, via a general beam facility. Further, data such as cross-section type, is more likely to be defined outside of the scope of the FEM reference model. It would logically be found in an A-E reference model.

Until it is shown that beam cross-sectional geometry is necessary for finite element analysis, it is the position of the committee that these data should not the defined in the FEM reference model. If the beam cross-sectional geometry is of interest, human readable text should be used to describe the cross-section via the FEM GEOMETRIC PROPERTY TEXT data entity (Entity 29), see Figure 110.

DECISION DATE: 11/10/86 by ISO Subgroup 2

ISSUE: FEM-19 FEM, FEA, and FEA CONTROL relationships.

INITIATION DATE: 04/16/86

INITIATOR: FEM Committee

STATUS: Unresolved

DESCRIPTION: The present STEP/PDES FEM planning model leaves the relationship between FEM, FEM ENVIRONMENT, FEA, FEA CONTROL, and FEA RESULTS data entities unclear. There are many key migrations which are incorrect and this is adding to the confusion. Before correcting the errors, it is important to reconsider what these entities are and how they interact so that their relationships are correctly reflected in the FEM reference model

BACKGROUND: The FEM data entity has nodes, elements, geometric properties, material properties, coordinate systems, and groups; i.e., all the basic information required to describe the model upon which several loading conditions and different analyses may be based. Obviously, a FEM entity can exist without any of the others mentioned here. Therefore its key attributes are simply FEM ID, as in the present reference model.

The FEM ENVIRONMENT data entity has loads, constraints, and load-case definitions. At present it is not possible to define the application of an environment in terms of geometric entities; i.e., surfac and edges. Even if this feature is provided in the future, it must st be possible to define the application in terms of nodes and elements as this is the most common way of applying loads and constraints. Therefore, the existence of a FEM ENVIRONMENT must be dependent upon the existence of a FEM and the FEM ID must be a key attribute. However, there are two ways that the key could migrate - directly from the FEM or indirectly from the FEA.

The FEA and FEA CONTROL data entities are basically doing a similar job; i.e., defining the analysis to be performed and can be more logically combined into one entity (in fact, they may have the same key attributes in the model). This entity should then be able to: refer to a model, select the load cases, describe the type of analysis to be performed, etc...

This entity is obviously dependent upon the existence of a model and the FEM ID should be a key attribute as shown in the STEP/PDES model, but should the FEM ENVIRONMENT ID be an attribute?

The FEA RESULTS data entity has displacements, stresses, strains, etc.... The results are dependent upon the existence of a FEM and a FEA so both of these IDs should be key attributes.

ISSUE: FEM-20 IDEF1X Failure to Model Ordered Lists.

INITIATION DATE: 11/02/86

INITIATOR: ISO Subgroup 2

STATUS: Resolved

DESCRIPTION: The failure of the IDEF1X modeling methodology to permit specification of an ordered list has been and continues to be an obstacle to the growth of the FEM reference model. What can be done about this situation?

Ordered lists should be allowed in the FEM reference model. There are many instances in the reference model were lists of parameters closely represent physical reality. For instance, a list of constrained displacements for several degrees of freedom, and a list of geometric properties at several distances along a beam. This type of list is easily defined using the EXPRESS language which the reference model will eventually be defined in, so why should we distort our data simply to accommodate an artifact of the IDEF1X modeling methodology?

EXPRESS can comprehensively represent ordered or unordered collections (sets) which may have fixed or varying sizes. For example, to represent an ordered collection of varying size, the construct LIST is used which states the upper and lower bounds on the list size (the upper bound may be unconstrained) and the type of each member of the list; e.g., REAL numbers or pointers to other entities.

RESOLUTION: After a discussion with the logical layer group about this problem, the following solution was reached. If FEM reference model constructs a data entity which is an artifact of the IDEF1X ordered list deficiency, then it will be carefully identified so that the offending data element will not be carried into the physical file schema.

DECISION DATE: 03/30/87 by ISO Subgroup 2

ISSUE: FEM–21 Tabular Material Data.

INITIATION DATE: 06/23/87

INITIATOR: ISO Subgroup 2

STATUS: Resolved

DESCRIPTION: The concept of tabular material property data needs to be captured by the IDEF1X data model. Presently, there is no mechanism to transmit a table of material properties.

RESOLUTION: Two new data entities, FEM MATERIAL TABLE and FEM MATERIAL TABLE INSTANCE, were defined to implement modeling of the tabular material data concept in a very general manner. These data entities represent the requirement that a material's properties sometimes vary with respect to independent variables. Typically, in structural mechanics problems where there is a wide range of temperature among a FEM's nodes, material properties are input as a tabular set of data with the independent variable being temperature and the dependent variables being the Young's modulus, Poisson's ratio, etc.... These data are interpolated with respect to temperature to determine what material property values to use.

The FEM MATERIAL TABLE and FEM MATERIAL TABLE INSTANCE are not restricted to tables with single independent variable The data construct presented here is sufficiently general to allow for tables that depend upon an arbitrary number of independent variables, because a FEM MATERIAL TABLE MATERIAL NUMBER attribute can have a value indicating that its corresponding FEM MATERIAL PROPERTY TYPE is, in fact, another FEM MATERIAL TABLE.

Figure 116 was modified to include the new FEM MATERIAL TABLE data entity. A new IDEF1X diagram, Figure 118, was created. This diagram depicts the relationship and cardinalities between the FEM MATERIAL PROPERTY, FEM MATERIAL TABLE, and FEM MATERIAL TABLE INSTANCE. The FEM MATERIAL TABLE and FEM MATERIAL TABLE INSTANCE data entities are numbered 61 and 62, respectively.

DECISION DATE: 06/23/87 by ISO Subgroup 2

ISSUE:	FEM-22 Material Problems for Dynamics problems.
INITIATION DATE:	06/22/87
INITIATOR:	ISO Subgroup 2
STATUS:	Resolved
DESCRIPTION:	When doing a static structural mechanics problem, the version 0.0 FEM IDEF1X data model did not permit the separation of material properties for static and dynamic problems. Data entities 38, 47, 51, and 52 need to be divided into new data entities by removing references to material properties used in dynamics.
RESOLUTION:	Two new data entities, FEM STRUCTURAL MASS DENSITY (Entity 59) and FEM STRUCTURAL DAMPING COEFFICIENT (Entity 60), were created to move the dynamics portion of the material properties out of entities 38, 47, 51, and 52. The effected IDEF1X diagrams, Figure 120, Figure 121, and Figure 125, were modified to reflect these requirements.
DECISION DATE:	06/22/87 by ISO Subgroup 2

ISSUE: FEM-23 Material Property Cardinality Logic.

INITIATION DATE: 06/23/87

INITIATOR: ISO Subgroup 2

STATUS: Resolved

DESCRIPTION: Version 0.0 of the FEM IDEF1X reference model used the concept of FEM ISOTROPIC MATERIAL TYPE, FEM ORTHOTROPIC MATERIAL TYPE, and FEM ANISOTROPIC MATERIAL TYPE. These types were used to select which material attributes were returned from a referenced material property. Thus, for example, either thermal or structural material properties could be defined for a single instance of FEM ISOTROPIC MATERIAL PROPERTY, but both groups of attributes could not exist within a single instance. Since a FEM ELEMENT can point to only one FEM GEOMETRIC PROPERTY (for constant properties) and, correspondingly, a single FEM GEOMETRIC PROPERTY is refers to by a single isotropic, orthotropic, or anisotropic material property, there was no way within the Version 0.0 reference model to call out both thermal and structural material properties for a specific FEM ELEMENT. Fix this problem.

RESOLUTION: The concept of FEM ISOTROPIC MATERIAL TYPE, FEM ORTHOTROPIC MATERIAL TYPE, and FEM ANISOTROPIC MATERIAL TYPE were removed from the FEM IDEF1X diagrams and the cardinality of all physical-quantity, material property data entities (38, 40, 41, 47, 48, 49, 50, 51, 52, 53, 59, and 60) were changed to have 0 or 1 occurrences with respect to the appropriate material property definition entity. The effected IDEF1X diagrams, Figure 120, Figure 121, and Figure 125, were modified to reflect these requirements.

DECISION DATE: 06/23/87 by ISO Subgroup 2

ISSUE: FEM-24 Thermal Expansion Property Seperation.

INITIATION DATE: 06/23/87

INITIATOR: ISO Subgroup 2

STATUS: Resolved

DESCRIPTION: Analogous to Issue FEM-22, the Version 0.0 reference model did not separate the thermal expansion data from other structural material data. These data should exist in separate data entities.

RESOLUTION: Four new data entities, FEM ISOTROPIC THERMAL EXPANSION (Entity 63), FEM ORTHOTROPIC THERMAL EXPANSION (Entity 64), FEM ANISOTROPIC 2D THERMAL EXPANSION (Entity 65), and FEM ANISOTROPIC 3D THERMAL EXPANSION (Entity 66), were created to move the thermal expansion data out of entities 38, 47, 51, and 52. The effected IDEF1X diagrams, Figure 120, Figure 121, and Figure 125, were modified to reflect these requirements.

DECISION DATE: 06/23/87 by ISO Subgroup 2

ISSUE: FEM-25 Real Keys.

INITIATION DATE: 06/23/87

INITIATOR: ISO Subgroup 2

STATUS: Resolved

DESCRIPTION: Real data types used as key attributes are problems for some data base management systems. Change all real key attributes to pairs of sequence numbers and values.

RESOLUTION: Examination of the FEM reference model revealed that there were only three data entities that contained key attributes which were defined with real data types. It was the consensus opinion of the committee that the concept of a sequence number should be inserted as the key attribute at each of these locations. Any real key attributes were moved to non-key attributes. Entities 33 [Figure 112], 45 [Figure 117], and 62 [Figure 118], were modified to reflect this decision. Three new key attributes, FEM MIXTURE VOLUME SEQUENCE NUMBER, FEM BEAM LENGTH FRACTION SEQUENCE NUMBER, and FEM MATERIAL TABLE INDEPENDENT VARIABLE SEQUENCE NUMBER, were added to the appropriate data entities and definitions for them were added to the attribute glossary.

This issue resolves the last issue present in [Shaw 1987a] that was n previously resolved, discarded, or documented as a FEM Issue.

DECISION DATE: 06/23/87 by ISO Subgroup 2

ISSUE:	FEM-26 Springs, Dampers and Beams.
INITIATION DATE:	10/12/87
INITIATOR:	FEM Committee
STATUS:	Resolved
DESCRIPTION:	The FEM BEAM GEOMETRIC PROPERTY IDEF1X reference model version 1.0 has serval problems associated with the concepts that it represents. This issue documents the changes made to this reference model to correct these problems.

RESOLUTION: It was the consensus opinion of the committee to divide the FEM BEAM GEOMETRIC PROPERTY IDEF1X diagram into two reference models, one for the beam geometric properties and the other for spring and damper geometric properties. On the new diagram, the spring and damper geometric properties are collected under a generic line geometric property entity.

These new diagram are now based upon the following model requirements:

1. The FEM IDEF1X model must include n-dimensional springs and dampers.
2. There can be only two nodes per spring or damper.
3. The degree of freedom sequence number are referenced to the node's output coordinate system.
4. The spring element which may be n-dimensional, can vary the spring coefficients with degree of freedom sequence number. Therefore, there can be as many spring coefficients for a spring element as there are nodal degree of freedom pairs.
5. There is only one spring coefficient for each degree of freedor pair in a spring element.
6. Statements (4) and (5) analogously apply to damper elements and damping coefficients.
7. Spring and damper data is not mutually exclusive.

Other information about this issue includes the following answered questions:

1. Since the FEM ELEMENT/NODE GEOMETRIC PROPERTY has no meaning for either the spring or damper elements defined in the current reference model, should there be a separate data path through the IDEF1X model to arrive at these entities?
   The same question applies to the two-noded beam element that may contain multiple stiffness intervals. It was decided to handle this situation with a rule. [Rule 12 (which was made obsolete by previous decisions in this issue) was replaced and now contains a statement that reflects this decision.]
2. Which coordinate system, either element or node, should be used as the coordinate system to reference degree of freedoms to?

It was decided for consistent that the node output coordinate syster be the reference for the degree of freedoms that is referred to by thes data.

SECTION 9: FEM INFORMATION MODEL

DECISION DATE: 10/12/87 by FEM Committee

ISSUE: FEM–27 Membrane Geometric Property.
INITIATION DATE: 10/12/87
INITIATOR: FEM Committee
STATUS: Resolved
DESCRIPTION: The FEM MEMBRANE GEOMETRIC PROPERTY IDEF1X diagram is a subset of the more general FEM SHELL GEOMETRIC PROPERTY. Should these two diagrams be combined into more general reference model?

RESOLUTION: It was the consensus opinion that the two reference models should be combined. The following notes document this decision and describe the changes required to insert membrane data into the new reference model.

1. Delete the FEM MEMBRANE GEOMETRIC PROPERTY reference model.
2. Inset FEM MEMBRANE GEOMETRIC PROPERTY entity (FEM-54) between the FEM GEOMETRIC PROPERTY TYPE line and the FEM SHELL GEOMETRIC PROPERTY entity (FEM-24). FEM MEMBRANE GEOMETRIC PROPERTY e tity (FEM-53) name is to change to FEM MEMBRANE/SHE GEOMETRIC PROPERTY.
3. There can be zero or one occuancies of FEM SHELL GEOMETRIC PROPERTY for each instance of FEM MEMBRANE/SHELL GEOMETRIC PROPERTY.
4. Remove the duplicated attributes in FEM SHELL GEOMETRIC PROPERTY.
5. Add a rule that states that the three material property foreign keys (non-key attributes) are optional. That is the user may not input a value for these which in itself is an important piece of information.

DECISION DATE: 10/12/87 by FEM Committee

ISSUE:	FEM-28 Fem Beam Model
INITIATION DATE:	04/01/87
INITIATOR:	ISO Subgroup 2
STATUS:	Resolved
DESCRIPTION:	The version 0.0 FEM BEAM GEOMETRIC PROPERTY reference model does not correctly model FEM beam element's geometric properties that vary with respect to length fractions. These types of properties cannot be properly defined via the FEM ELEMENT/NODE GEOMETRIC PROPERTY because these entities exist only at an element's nodes (thus having a specific length fraction). A length fraction is arbitrary by definition and these properties may therefore occur at any length fraction along the beam finite element.

RESOLUTION: (Open is fixing issue which is unresolved)
The basis for this resolution can be found in [Leal 1987a]. Restated here for clarity and consistency with the rest of the specification. The FEM beam element modeled for the FEM BEAM GEOMETRIC PROPERTY is assembled from the following components:

1. An optional release at the nodes of a beam is required. One or more of the nodal degree of freedoms can be released. The release coordinate system will be the nodal output coordinate system.
2. An optional beam connector (fixing) is required. (at the present time, further clarification is being sought about how this should actually be modeled. No FEM model reference is made, as yet, to these data.)
3. An optional offset is required between the position of the beam connector (fixing) with respect to a node. This offset is a vector relative to a node in the node's definition coordinate system (????).
4. One or optionally many flexure intervals are required. These intervals can have discontinuous geometric properties at the i terval boundaries. Linear interpolation is used to calculate variations of geometric properties with an interval. Intervals must be continuous.

A schematic diagram of this beam model is shown in figure ??.
Other assumptions:

- Beam geometric properties are given with respect to the beam's geometric property coordinate system. The beam's geometric property coordinate system (see the attribute FEM ELEMENT GEOMETRIC PROPERTY COORDINATE SYSTEM) is oriented with is local x-axis being a straight line between the beam's fixings. The other mutually orthogonal directions are chosen for the convenient definition of the values instanced in a FEM BEAM PROPERTY DAA entity (FEM-33). (These data include the cross-sectional area, the second moments of area, etc...)
- Beam length fraction are calculated with respect to the distance between beam fixings. These values of the beam length fraction at fixing/1 and fixing/2 are 0,0 and 1,0, respectively. The first flexure interval must begin at length fraction of 0,0 and the last flexure interval in the sequence must end at length fraction of 1,0.

CONTINUED:

- A Flexure interval may have either constant of linearly varying section geometric properties, See Figure ??. Interval boundaries may be discontinuous.
- Material properties do not vary within flexure intervals.
- Material properties data may be different for different flexure sections.
- Warping coefficient data is required only at the ends of the beam.
- Optional beam stress recovery information can be specified in conjunction with flexure interval information.

DECISION DATE: 10/12/87 by FEM Committee

NOTES:

1. Also, included in the new IDEF1X reference model is a reference to a standard section. We expect this reference to connect with some, as yet undefined, set of tabular data probably located in the AEC reference model. Issue FEM-18 has further discussion on this concept.

ISSUE: FEM–29 Preventing Fatal Recursive Material Id Reference.
INITIATION DATE: 03/31/88
INITIATOR: FEM Committee
STATUS: Resolved
DESCRIPTION: FEM COMPOSITE MATERIAL PROPERTY entity (FEM-35) can be one of three different type of composite material. Each composite material type has request for two or more materials in its' make-up. The current model allows for a composite material to request itself.

RESOLUTION: Rule FEM/R-15 was created in order to avoid a fatal recursion. That rule is as follows:
No material property references in a FEM COMPOSITE MATERIAL PROPERTY can be to itself.
DECISION DATE: 03/31/88 by FEM Committee

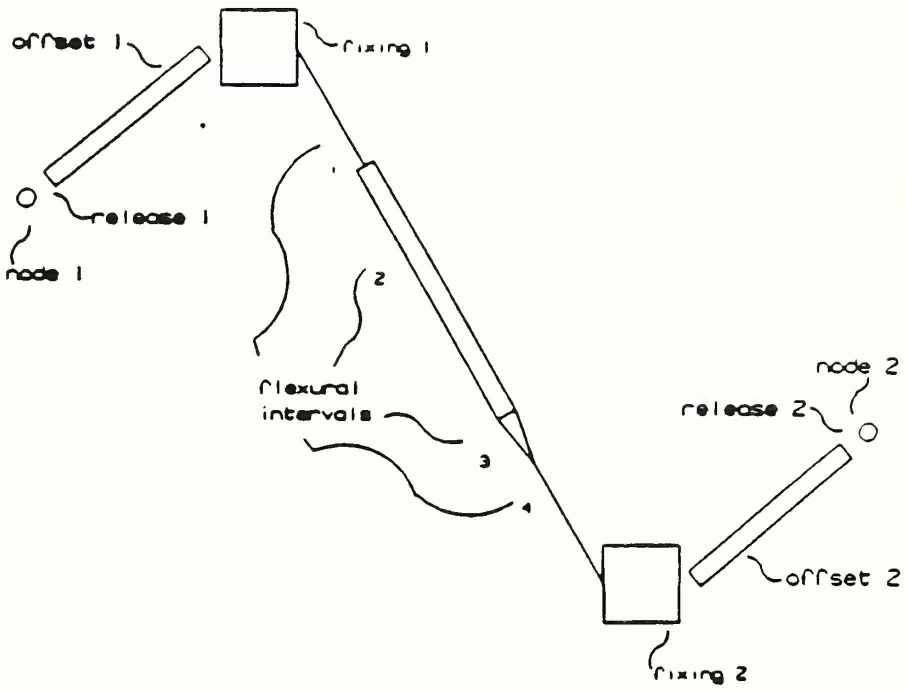


Figure D-146: Beam Schematic Diagram

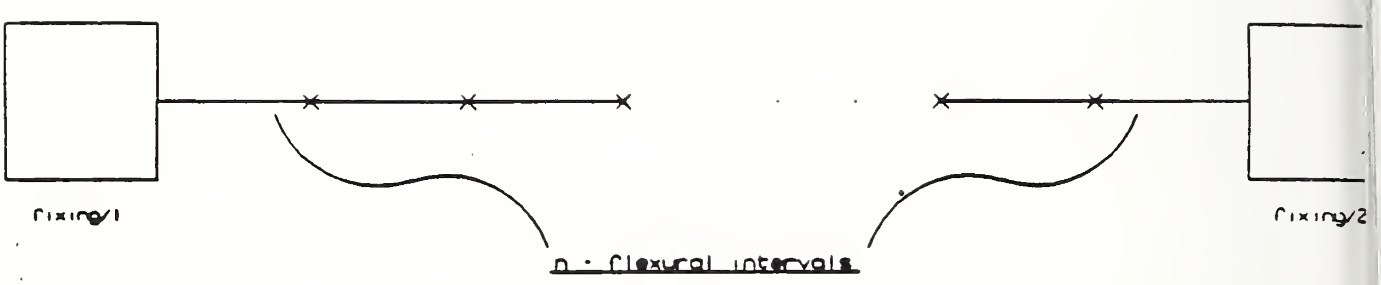


Figure D-147: Flexural Intervals

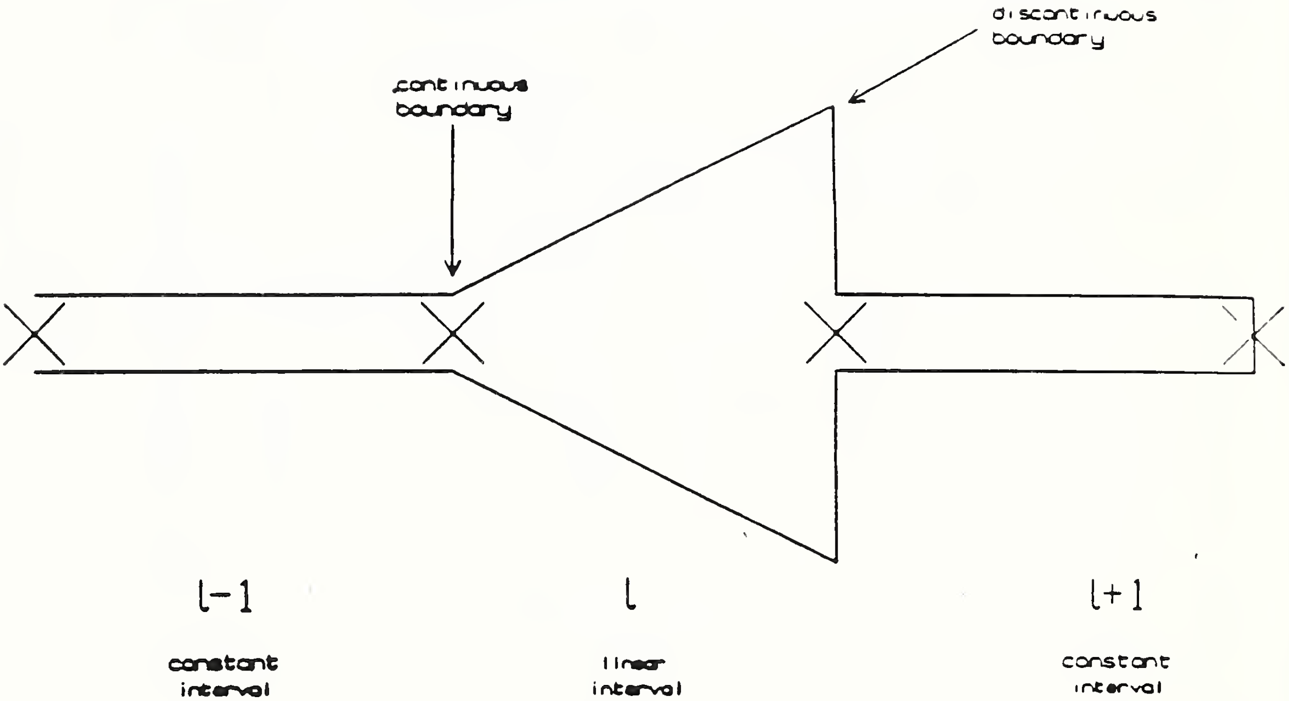


Figure D-148: Flexural Interval Types

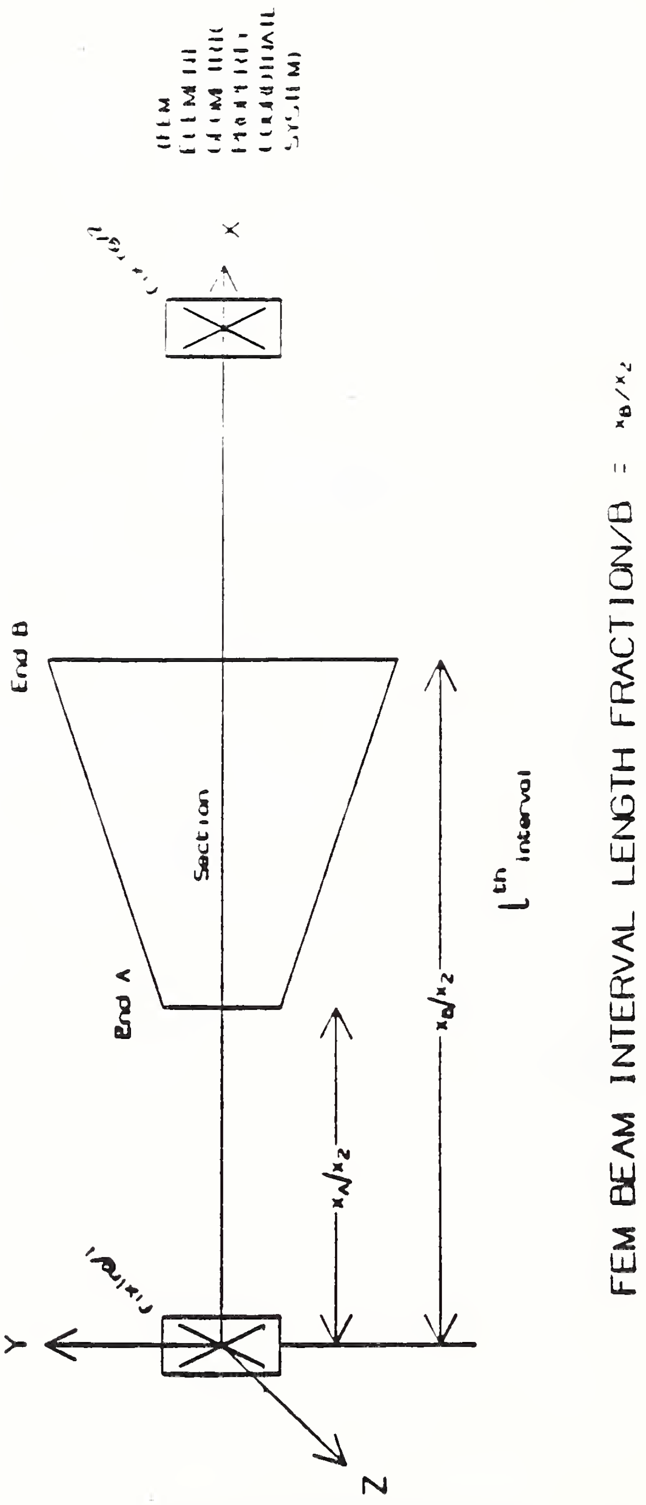


Figure D-149: Flexural Interval End Definitions

ISSUE: FEM-30 FEM Material Allowables.
INITIATION DATE: 01/14/88
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: There should be a data type for FEM material allowables that is related to FEM MATERIAL PROPERTY entity (FEM-12).

RESOLUTION: The types of allowables that should be captured are for example : Tensile and compressive allowables for each component of a elastic matrix.

NOTES:

- Figure out whether to hang off of material type or off of each sub-type.
- A way to describe a variable number of allowables in needed.
- ? A sub-type of each sub-type or what????

ISSUE:	FEM-31 Material Property Coordinate System.
INITIATION DATE:	01/14/88
INITIATOR:	FEM Committee
STATUS:	Open
DESCRIPTION:	There should be a pointer to a coordinate system from FEM MATERIAL PROPERTY entity (FEM-12) in order to align the material directions.

NOTES:

- Figure out if there is a need
- Assign a pointer mechanism
- Deal with the "over-riding" of element material coordinate system attribute
- Could the element material coordinate system attribute handle this also.
- Discussion about need for material property direction vs material angle in a FEM element. Defines position for correct location of the element material property data.

ISSUE: FEM-32 Point Damping Coefficient.
INITIATION DATE: 01/14/88
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: Version 1.1 is missing a point damping element.

RESOLUTION: Point damping is a feature that is commonly used by dynamic analyst in aerospace. Therefore the model should be correct to allow for point damping.

ISSUE:	FEM-33 Time Stamp Relationship Between Product Item and FEM.
INITIATION DATE:	01/14/88
INITIATOR:	FEM Committee
STATUS:	Open
DESCRIPTION:	A question was raised about the time stamp relationship between Product Item and FEM.
RESOLUTION:	A question was raised about the time stamp relationship between Product Item and FEM. The following are some of the question raised:

1. Should both items, FEM and Product Item, be dated? The FEM committee determined early on that the FEM needs to be dated and from that decision, we would like to see Product Item dated also.

2. This issue should be passed to the mechanical product committee that owns the PSCM model in order that this issue can be properly addressed.

3. This issue should also be pass onto the integration committee ir order to see how (and if)this is addressed there.

ISSUE: FEM-34 Mesh Generation Entities.
INITIATION DATE: 01/14/88
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: How and should mesh generation entities be related to the Product Item geometric entities?

RESOLUTION: How and should mesh generation entities be related to the Product Item geometric entities? we are interested in connecting FEM nodes and element to the following geometry type (by commonly called names):

- Points
- Lines/curves
- Patch/surface/face
- Solid/volume/shell

This issue is placed on hold until the integration committee completes its current work of how to relate non-geometry items and geometry. Also they are working on a method of handling the connection of different types of geometry so that the non-geometric model only point to one entity in order to get to the provided (underlining) geometry.

ISSUE: FEM-35 Part Geometry and FEM geometry are not Identical.
INITIATION DATE: 01/14/88
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: How should Product Item geometry and mesh-generation entities that are related but do not have identical be handled?

RESOLUTION: Example of the need are following:

1. The idealization of a web in a airplane by a FEM membrane shear-panel element that requiring geometric data at the centerline of the web's thickness.
2. Construction lines, curves, surfaces, and solids that were created in order to aid in the generation of nodes and elements which are later connected directly.

NOTES:

- Obviously existing geometric entities should be used.
- How and should FEM entities be related? (See Issue FEM-34)
- This issue should be pass on to the integration and geometr committees.

ISSUE:	FEM-36 FEM Material and Geometric Property Libraries.
INITIATION DATE:	03/30/88
INITIATOR:	FEM Committee
STATUS:	Open
DESCRIPTION:	The FEM material and geometric property's existence need not necessarily depend on the existence of a FEM. This would allow for the possibility of creating libraries of these properties. These library can be in turned be shared by several FEM's.

ISSUE: FEM-37 FEM Geometric Property Uniqueness.
INITIATION DATE: 01/14/88
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: Some analysis codes allow for the same geometric property number (id) to be used by several different type of geometric properties. Version 1.1 does not allow for this.

RESOLUTION: Some analysis codes allow for the same geometric property number (id) to be used by several different type of geometric properties. For example, the geometric property number of 1 could be used in MSC/NASTRAN by both the beam geometric property and shell geometric property.
One suggested solution is to added to the geometric property entity the element type. Other is to make the FEM GEOMETRY PROPERTY TYPE attribute as key attribute instead of a non-key attribute.

ISSUE: FEM-38 Text for FEM, Material, Control, Groups and Environment.
INITIATION DATE: 10/12/87
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: Need to be able to point to a text entity for the following entities: Groups, Materials, Environment, Controls, and FEM in order to store related information.

ISSUE:	FEM-39 Node Ordering for FEM Elements.
INITIATION DATE:	09/14/88
INITIATOR:	FEM Committee
STATUS:	Open
DESCRIPTION:	There is a need to define for the varies FEM element types the nodal order. For this nodal order is used to define the geometric shape of the FEM element. Important item that influence the nodal ordering are the element's shape type (line, prism, etc.) and the element's order number.
RESOLUTION:	The current solution arrived at in the 1988 Denver meeting was to define the nodal ordering for following and leave the rest for future work:

Element Type	Reference Order	Figure
Line	0 and 1	129
	2	129
	3 . . .	129
Triangle	0 and 1	130
	2	130
Quadrilateral	0 and 1	131
	2	131
Tetrahedron	0 and 1	132
	2	132
Wedge	0 and 1	133
	2	133
Prism	0 and 1	135
	2	135
Hexahedron	0 and 1	137
	2	137

NOTES: The FEM committee did a light review of varies FEM codes and how they do nodal ordering and their distribution of nodes for their varies element type. It was noted that varies element types and for element orders 0, 1 and 2 there was good agreement between FEM codes on their distribution of nodes, but not on how they order them.

Therefore the FEM committee chose to use the common distribution of nodes that was found and is documented in figures 129 through 137 (see the above table for details). The FEM committee chose an nodal ordering for there was not a common and that is also documented in figures 129 through 137 and for detail see those figure in entity FEM CONNECTIVITY [Entity FEM-6].

The FEM committee had problems defining the distribution of nodes for the high order element and that this time those were not defined and left for future work by the FEM committee.

9.7 Problem Areas and Unimplemented Ideas

FEA Control Solution Ordering

If a solution ordering exists simultaneously with both a case control and a FE control, then the solution ordering specified with the case control takes precedence.

Date Entered: 10/22/86 by FEM Committee

FEA Dynamic Condensation

If a dynamic condensation exists simultaneously with both a case control and a FE control, then the dynamic condensation specified with the case control takes precedence.

Date Entered: 10/22/86 by FEM Committee

Rigid Elements

Is it sufficient to assume that all forms of rigid finite elements can be reduced to simple forms of multipoint constraint equations and therefore all forms of rigid finite elements can be represented in the FEM data model as FEM MULTIPOINT CONSTRAINTS?

Date Entered: 10/22/86 by FEM Committee

Gaps and Cables

Can gap and cable finite elements be handled sufficiently with nonlinear material properties or is it necessary to indicate there geometric nonlinearity in another manner? Will a business rule stipulating that rigid elements be handled as multipoint constraints be satisfactory?

Date Entered: 10/22/86 by FEM Committee

Initial Velocities and Accelerations

How can initial velocities and accelerations be implemented into the FEM data model in a generic manner?

Date Entered: 10/23/86 by FEM Committee

Shock Spectra and Forcing Functions

Where should shock spectra and forcing functions be inserted into the FEM data model?

Date Entered: 10/23/86 by FEM Committee

Element Local Coordinate Systems

If we need to define finite element local coordinate systems, how do we do it?

Date Entered: 04/01/87 by ISO Subgroup 2

Bar or Rod Elements

How do we distinguish between beam finite elements and bars or rods which have no bending stiffness (i.e., there second moments of area are approximately zero)?

Date Entered: 04/01/87 by ISO Subgroup 2

Beam Pin Flags

Beam pin flag data needs more work. It does not make sense where is currently shows up on the reference model.

Date Entered: 04/01/87 by ISO Subgroup 2

Pipe Elements

How do we model pipe elements and beam elements with more than 2 nodes?

Date Entered: 04/01/87 by ISO Subgroup 2

Failure Criteria

How do we model yield and ultimate stresses?
Should we model failure criteria?

Date Entered: 01/14/88 by FEM Committee

General Stiffness Element

How do we model a general stiffness element?

Date Entered: 01/14/88 by FEM Committee

9.8 FEM Complete SML Model

The following is a listing of the FEM SML model.

```
 VIEW fem-model
 GENERAL INFO
 SHORT NAME fem-model
 AUTHOR Richard Brooks
 DESCRIPTION this is the FEM data model for version 1.1
 CREATION DATE 04/19/87
*
*       This is the FEM model.
*       JANUS (SML) input
*       April 88
*
 STATUS develop
 LEVEL kb
 STRUCTURE

ENTITY  FEM
 KEY
        FEM-ID
 ENDKEY
        FEM-AUTHOR
        FEM-CREATING-SOFTWARE
        FEM-CREATION-DATE
        FEM-CREATION-TIME
        FEM-DESCRIPTOR
        FEM-TARGETED-ANALYSIS-CODE
        UNITS "has"
        ROLE fem-units-id FOR units-id
ENDENTITY

ENTITY PRODUCT-ITEM-fem
 KEY
        FEM "models"
        PRODUCT-ITEM "defines"
 ENDKEY
ENDENTITY

ENTITY  PRODUCT-ITEM
 KEY
        PRODUCT-ITEM-ID
 ENDKEY
```

```
ENDENTITY

ENTITY  UNITS
 KEY
        UNITS-ID
 ENDKEY
        LENGTH-UNIT
        MASS-UNIT
        TIME-UNIT
        ELECTRIC-CURRENT-UNIT
        THERMODYNAMIC-TEMPERATURE-UNIT
        AMOUNT-OF-SUBSTANCE-UNIT
        luminous-intensity-unit
        PLANE-ANGLE-UNIT
        SOLID-ANGLE-UNIT
ENDENTITY

ENTITY  FEM-APPROVAL
 KEY
        FEM-APPROVAL-ID
        FEM "is approved by"
 ENDKEY
        FEM-APPROVAL-DATE
        FEM-APPROVER-NAME
ENDENTITY

ENTITY  FEA-CONTROL
 KEY
        FEA-CONTROL-ID
 ENDKEY
        FEA-ANALYSIS-ASSUMPTION
        FEA-ANALYSIS-TYPE
        FEA-TARGETED-ANALSIS-CODE
        UNITS "HAS"
        role FEA-CONTROL-UNITS-ID for UNITS-ID
ENDENTITY

ENTITY  FEA-CONTROL-APPROVAL
 KEY
        FEA-CONTROL-APPROVAL-ID
        FEA-CONTROL "is approved by"
 ENDKEY
        FEA-CONTROL-APPROVAL-DATE
        FEA-CONTROL-APPROVER-NAME
ENDENTITY

ENTITY  FEA
 KEY
        FEA-ID
        FEM "HAS"
        FEA-CONTROL "HAS"
 ENDKEY
        FEA-ACTUAL-ANALYSIS-CODE
        FEA-AUTHOR
        UNITS "HAS"
        role FEA-UNITS-ID for UNITS-ID
ENDENTITY

ENTITY  FEM-ELEMENT
 KEY
        FEM "will have" P
        FEM-ELEMENT-NUMBER
 ENDKEY
        FEM-COORDINATE-SYSTEM "HAS"
        role FEM-ELEMENT-geometric-property-COORDINATE-SYSTEM-number for
             FEM-COORDINATE-SYSTEM-NUMBER
        FEM-COORDINATE-SYSTEM "HAS"
        role FEM-ELEMENT-LOCAL-COORDINATE-SYSTEM-number for
             FEM-COORDINATE-SYSTEM-NUMBER
        FEM-COORDINATE-SYSTEM "HAS"
        role FEM-ELEMENT-MATERIAL-PROPERTY-COORDINATE-SYSTEM-number for
             FEM-COORDINATE-SYSTEM-NUMBER
        FEM-ELEMENT-PURPOSE-DESCRIPTOR
        FEM-ELEMENT-ORDER-NUMBER
        FEM-ELEMENT-SHAPE-NUMBER
ENDENTITY

ENTITY  FEM-NODE
 KEY
        FEM "will have" p
        FEM-NODE-NUMBER
 ENDKEY
        FEM-NODE-COORDINATE-1
        FEM-NODE-COORDINATE-2
        FEM-NODE-COORDINATE-3
        FEM-COORDINATE-SYSTEM "HAS"
        role FEM-NODE-DEFINITION-COORDINATE-SYSTEM-number for
```

```
                FEM-COORDINATE-SYSTEM-number
          FEM-COORDINATE-SYSTEM "HAS"
          role FEM-NODE-OUTPUT-COORDINATE-SYSTEM-number for
                FEM-COORDINATE-SYSTEM-number
ENDENTITY

ENTITY FEM-CONNECTIVITY
 KEY
          FEM-ELEMENT "will have" P
          FEM-NODE-SEQUENCE-NUMBER
 ENDKEY
          FEM-NODE "is member of"
ENDENTITY

ENTITY  FEM-COORDINATE-SYSTEM
 KEY
          FEM "HAS"
          FEM-COORDINATE-SYSTEM-number
 ENDKEY
 GENERALIZATION BY FEM-COORDINATE-SYSTEM-use OF
          fem-coordinate-master-system
          fem-coordinate-definition-system
 ENDGENERALIZATION
          FEM-COORDINATE-system-type
ENDENTITY

ENTITY  fem-coordinate-master-system
 CATEGORY BY fem-coordinate-system-use
 OF fem-coordinate-system
ENDENTITY

ENTITY  fem-coordinate-definition-system
 CATEGORY BY fem-coordinate-system-use
 OF fem-coordinate-system
ENDENTITY

ENTITY  fem-coordinate-transform
 KEY
          fem-coordinate-master-system "has"
          ROLE fem-coordinate-master-system-number FOR
                fem-coordinate-system-number
          fem-coordinate-definition-system "has"
          ROLE fem-coordinate-definition-system-number FOR
                fem-coordinate-system-number
 ENDKEY
        fem-coordinate-transformation-matrix
ENDENTITY

ENTITY  FEM-GROUP
 KEY
        FEM "HAS"
        FEM-GROUP-number
 ENDKEY
        fem-group-name
ENDENTITY

ENTITY  FEM-NODE-GROUP
 KEY
        FEM-GROUP "HAS"
        FEM-NODE "HAS"
 ENDKEY
ENDENTITY

ENTITY  FEM-ELEMENT-GROUP
 KEY
        FEM-GROUP "HAS"
        FEM-ELEMENT "HAS"
 ENDKEY
ENDENTITY

ENTITY  FEM-GEOMETRIC-PROPERTY
 KEY
        FEM "HAS"
        FEM-GEOMETRIC-PROPERTY-number
 ENDKEY
 generalization by FEM-GEOMETRIC-PROPERTY-TYPE
 of
        fem-point-geometric-property
        fem-beam-geometric-property
        fem-line-geometric-property
        fem-membrane-shell-geometric-property
        fem-solid-geometric-property
 ENDGENERALIZATION
ENDENTITY

ENTITY FEM-GEOMETRIC-PROPERTY-TEXT
 KEY
        FEM-GEOMETRIC-PROPERTY "HAS" Z
 ENDKEY
        FEM-GEOMETRIC-PROPERTY-TEXT
ENDENTITY

ENTITY FEM-GEOMETRIC-PROPERTY-INTEGRATION-ORDER
 KEY
        FEM-GEOMETRIC-PROPERTY "HAS" Z
 ENDKEY
        FEM-INTEGRATION-ORDER
ENDENTITY

ENTITY  fem-element-geometric-property
 KEY
        fem-element "has" z
        fem-geometric-property "has"
 ENDKEY
ENDENTITY

ENTITY  fem-element-node-geometric-property
 KEY
        fem-connectivity "has" z
        fem-geometric-property "has"
 ENDKEY
ENDENTITY

ENTITY  fem-point-geometric-property
 CATEGORY BY fem-geometric-property-type
 OF     fem-geometric-property
  point-mass-half-6x6-matrix
ENDENTITY


ENTITY fem-line-geometric-property
 CATEGORY BY fem-geometric-property-type
 OF     fem-geometric-property
 GENERALIZATION BY fem-line-geometric-property-type
 OF     fem-spring-geometric-property
   .    fem-damper-geometric-property
 ENDGENERALIZATION
 ENDENTITY

ENTITY fem-spring-geometric-property
 CATEGORY BY fem-line-geometric-property-type
 OF     fem-line-geometric-property
```

```
ENDENTITY


ENTITY fem-damper-geometric-property
 CATEGORY BY fem-line-geometric-property-type
 OF      fem-line-geometric-property
ENDENTITY

ENTITY   fem-spring-property
 KEY
         fem-spring-geometric-property "has"
         fem-spring-dof-sequence-no-1
         fem-spring-dof-sequence-no-2
  ENDKEY
         fem-spring-coefficient
         fem-stress-recovery-coefficient
ENDENTITY

ENTITY  fem-damper-property
 KEY
         fem-damper-geometric-property "has"
         fem-damper-dof-sequence-no-1
         fem-damper-dof-sequence-no-2
  ENDKEY
         fem-damper-coefficient
ENDENTITY

ENTITY  fem-beam-geometric-property
 CATEGORY BY fem-geometric-property-type
 OF      fem-geometric-property
ENDENTITY

ENTITY  fem-beam-warping-data
 KEY
         fem-beam-geometric-property "has"
 ENDKEY
         fem-beam-warping-coefficients-1
         fem-beam-warping-coefficients-2
ENDENTITY

ENTITY  fem-beam-offset-vectors
 KEY
         fem-beam-geometric-property "has"
 ENDKEY
```

```
        fem-beam-offset-vector-1
        fem-beam-offset-vector-2
ENDENTITY

ENTITY  fem-beam-pin-data
 KEY
        fem-beam-geometric-property "has"
 ENDKEY
        fem-beam-pin-array-1
        fem-beam-pin-array-2
ENDENTITY

ENTITY  fem-beam-interval
 KEY
        fem-beam-geometric-property "has"
        fem-beam-interval-number
 ENDKEY
        fem-beam-length-fraction-b
        fem-beam-section "has"
        ROLE fem-beam-section-a FOR fem-beam-section-number
        fem-beam-section "has"
        ROLE fem-beam-section-b FOR fem-beam-section-number
        fem-material-property "has"
ENDENTITY

ENTITY  fem-beam-section
 KEY
        fem "has"
        fem-beam-section-number
 ENDKEY
 GENERALIZATION BY fem-beam-section-type
 OF     fem-beam-standard-section
        fem-beam-property-data
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-beam-standard-section
 CATEGORY BY fem-beam-section-type
 OF     fem-beam-section
        standard-section "has"
ENDENTITY

ENTITY  fem-beam-property-data
 CATEGORY BY fem-beam-section-type
 OF      fem-beam-section
         fem-beam-cross-sectional-area
         fem-beam-second-moments-of-area
         fem-beam-polars-moment
         fem-beam-shear-area
         fem-beam-shear-relief-coefficients
         fem-beam-non-structural-mass
         fem-beam-centroid-offset
         fem-beam-shear-centre-offset
         fem-beam-non-structural-mass-offset
ENDENTITY

ENTITY  fem-beam-stress-recovery-coefficient
 KEY
         fem-beam-section "has"
         fem-beam-stress-recovery-coefficient-number
 ENDKEY
         fem-beam-local-y-coordinate
         fem-beam-local-z-coordinate
ENDENTITY

ENTITY  standard-section
 KEY
         standard-section-id
 ENDKEY
ENDENTITY

ENTITY  fem-membrane-shell-geometric-property
 CATEGORY BY fem-geometric-property-type
 OF      fem-geometric-property
         fem-material-property "has"
         ROLE fem-membrane-material-property-number FOR
              fem-material-property-number
         fem-membrane-non-structural-mass
         fem-membrane-thickness
ENDENTITY

ENTITY  fem-shell-geometric-property
 KEY
         fem-membrane-shell-geometric-property "has" z
 ENDKEY
         fem-material-property "has"
         ROLE fem-shell-bending-material-property-number FOR
              fem-material-property-number
```

```
        fem-material-property "has"
        ROLE fem-shell-shear-material-property-number FOR
             fem-material-property-number
        fem-material-property "has"
        ROLE fem-shell-membrane-bending-material-property-number for
             fem-material-property-number
        fem-shell-second-moment-of-area
        fem-shell-shear-thickness
        fem-shell-z-offset
  stress-recovery-coefficient
ENDENTITY

ENTITY  fem-shell-stress-recovery-coefficient
 KEY
        fem-shell-geometric-property "has"
        fem-shell-stress-recovery-coefficient-number
 ENDKEY
        fem-shell-stress-recovery-coefficient
ENDENTITY

ENTITY  fem-solid-geometric-property
 CATEGORY BY fem-geometric-property-type
 OF     fem-geometric-property
        fem-material-property "has"
        ROLE fem-solid-material-property-number FOR
             fem-material-property-number
ENDENTITY

ENTITY  fem-material-property
 KEY
        fem "has"
        fem-material-property-number
 ENDKEY
 GENERALIZATION BY fem-material-property-type
 OF     fem-composite-material-property
        fem-homogeneous-material-property
        fem-material-table
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-homogeneous-material-property
 CATEGORY BY fem-material-property-type
 OF     fem-material-property
 GENERALIZATION BY fem-homogeneous-material-property-type
 OF      fem-isotropic-material-property
         fem-orthotropic-material-property
         fem-anisotropic-material-property
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-structural-mass-density
 KEY
         fem-homogeneous-material-property "has" z
 ENDKEY
         fem-structural-mass-density-value
ENDENTITY

ENTITY  fem-structural-damping-coefficient
 KEY
         fem-homogeneous-material-property "has" z
 ENDKEY
         fem-structural-damping-coefficient-value
ENDENTITY

ENTITY  fem-isotropic-material-property
 CATEGORY BY fem-homogeneous-material-property-type
 OF      fem-homogeneous-material-property
ENDENTITY

ENTITY  fem-isotrpic-thermal-material-property
 KEY
         fem-isotropic-material-property "has" z
 ENDKEY
         fem-thermal-conductivity-coefficient
         fem-specific-heat-capacity
         fem-convective-film-coefficient
ENDENTITY

ENTITY  fem-isotropic-structural-material-property
 KEY
         fem-isotropic-material-property "has" z
 ENDKEY
         fem-isotropic-Youngs-modulus
         fem-isotropic-poissons-ratio
         fem-isotropic-shear-modulus
ENDENTITY

ENTITY  fem-isotropic-thermal-expansion
```

```
 KEY
        fem-isotropic-material-property "has" z
 ENDKEY
        fem-isotropic-thermal-expansion-coefficient
        fem-isotropic-thermal-expansion-reference-temperature
ENDENTITY

ENTITY  fem-orthotropic-material-property
 CATEGORY BY fem-homogeneous-material-property-type
 OF     fem-homogeneous-material-property
ENDENTITY

ENTITY  fem-orthotropic-thermal-material-property
 KEY
        fem-orthotropic-material-property "has" z
 ENDKEY
 GENERALIZATION BY fem-orthotropic-thermal-material-property-type
 OF     fem-orthotropic-thermal-2d-material-property
        fem-orthotropic-thermal-3d-material-property
 ENDGENERALIZATION
        fem-orthotropic-thermal-in-plane-thermal-conductivities
        fem-orthotropic-thermal-heat-capacity
        fem-orthotropic-thermal-convective-film-coefficient
ENDENTITY

ENTITY  fem-orthotropic-thermal-2d-material-property
 CATEGORY BY fem-orthotropic-thermal-material-property-type
 OF     fem-orthotropic-thermal-material-property
ENDENTITY

ENTITY  fem-orthotropic-thermal-3d-material-property
 CATEGORY BY fem-orthotropic-thermal-material-property-type
 OF     fem-orthotropic-thermal-material-property
        fem-orthotropic-thermal-out-plane-thermal-conductivity
ENDENTITY

ENTITY  fem-orthotropic-structural-material-property
 KEY
        fem-orthotropic-material-property "has" z
 ENDKEY
 GENERALIZATION BY fem-orthotropic-structural-material-property-type
 OF     fem-orthotropic-structural-2d-material-property
        fem-orthotropic-structural-3d-material-property
 ENDGENERALIZATION
```

```
        fem-orthotrpic-structural-in-plane-poissons-ratio
        fem-orthotrpic-structural-in-plane-shear-modulus
        fem-orthotrpic-structural-in-plane-youngs-moduli
ENDENTITY

ENTITY  fem-orthotropic-structural-2d-material-property
 CATEGORY BY fem-orthotropic-structural-material-property-type
 OF      fem-orthotropic-structural-material-property
ENDENTITY

ENTITY  fem-orthotropic-structural-3d-material-property
 CATEGORY BY fem-orthotropic-structural-material-property-type
 OF      fem-orthotropic-structural-material-property
        fem-orthotrpic-structural-out-plane-poissons-ratio
        fem-orthotrpic-structural-out-plane-shear-modulus
        fem-orthotrpic-structural-out-plane-youngs-moduli
ENDENTITY

ENTITY  fem-orthotropic-thermal-expansion
 KEY
        fem-orthotropic-material-property "has" z
 ENDKEY
 GENERALIZATION BY fem-orthotropic-thermal-expansion-type
 OF      fem-orthotropic-2d-thermal-expansion
        fem-orthotropic-3d-thermal-expansion
 ENDGENERALIZATION
        fem-orthotrpic-thermal-in-plane-expansion-coefficients
        fem-orthotrpic-thermal-expansion-reference-temperature
ENDENTITY

ENTITY  fem-orthotropic-2d-thermal-expansion
 CATEGORY BY fem-orthotropic-thermal-expansion-type
 OF      fem-orthotropic-thermal-expansion
ENDENTITY

ENTITY  fem-orthotropic-3d-thermal-expansion
 CATEGORY BY fem-orthotropic-thermal-expansion-type
 OF      fem-orthotropic-thermal-expansion
        fem-orthotrpic-out-plane-thermal-expansion-coefficients
ENDENTITY

ENTITY  fem-anisotropic-material-property
 CATEGORY BY fem-homogeneous-material-property-type
 OF      fem-homogeneous-material-property
 GENERALIZATION BY fem-anisotropic-material-property-type
 OF      fem-anisotropic-thermal-expansion
         fem-anisotropic-thermal-material-property
         fem-anisotropic-structural-material-property
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-anisotropic-thermal-material-property
 CATEGORY BY fem-anisotropic-material-property-type
 OF      fem-anisotropic-material-property
 GENERALIZATION BY fem-anisotropic-thermal-material-property-type
 OF      fem-anisotropic-thermal-2d-material-property
         fem-anisotropic-thermal-3d-material-property
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-anisotropic-thermal-2d-material-property
 CATEGORY BY fem-anisotropic-thermal-material-property-type
 OF      fem-anisotropic-thermal-material-property
         fem-anisotropic-thermal-2d-conductivity-matrix
         fem-anisotropic-thermal-2d-heat-capacity
ENDENTITY

ENTITY  fem-anisotropic-thermal-3d-material-property
 CATEGORY BY fem-anisotropic-thermal-material-property-type
 OF      fem-anisotropic-thermal-material-property
         fem-anisotropic-thermal-3d-conductivity-matrix
         fem-anisotropic-thermal-3d-heat-capacity
ENDENTITY

ENTITY  fem-anisotropic-structural-material-property
 CATEGORY BY fem-anisotropic-material-property-type
 OF      fem-anisotropic-material-property
 GENERALIZATION BY fem-anisotropic-structural-material-property-type
 OF      fem-anisotropic-structural-2d-material-property
         fem-anisotropic-structural-3d-material-property
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-anisotropic-structural-2d-material-property
 CATEGORY BY fem-anisotropic-structural-material-property-type
 OF      fem-anisotropic-structural-material-property
         fem-anisotropic-structural-2d-matrix
ENDENTITY
```

```
ENTITY  fem-anisotropic-structural-3d-material-property
 CATEGORY BY fem-anisotropic-structural-material-property-type
 OF      fem-anisotropic-structural-material-property
         fem-anisotropic-structural-3d-matrix
ENDENTITY

ENTITY  fem-anisotropic-thermal-expansion
 CATEGORY BY fem-anisotropic-material-property-type
 OF      fem-anisotropic-material-property
 GENERALIZATION BY fem-anisotropic-thermal-expansion-type
 OF      fem-anisotropic-2d-thermal-expansion
         fem-anisotropic-3d-thermal-expansion
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-anisotropic-2d-thermal-expansion
 CATEGORY BY fem-anisotropic-thermal-expansion-type
 OF      fem-anisotropic-thermal-expansion
         fem-anisotropic-2d-thermal-expansion-coefficients
         fem-anisotropic-2d-thermal-expansion-reference-temperature
ENDENTITY

ENTITY  fem-anisotropic-3d-thermal-expansion
 CATEGORY BY fem-anisotropic-thermal-expansion-type
 OF      fem-anisotropic-thermal-expansion
         fem-anisotropic-3d-thermal-expansion-coefficients
         fem-anisotropic-3d-thermal-expansion-reference-temperature
ENDENTITY

ENTITY  fem-composite-material-property
 CATEGORY BY fem-material-property-type
 OF      fem-material-property
 GENERALIZATION BY fem-composite-material-type
 OF      fem-halpin-tsai-material-property
         fem-laminate-composite-material-property
         fem-mixture-composite-material-property
 ENDGENERALIZATION
ENDENTITY

ENTITY  fem-mixture-composite-material-property
 CATEGORY BY fem-composite-material-type
 OF      fem-composite-material-property
ENDENTITY
```

```
ENTITY  fem-mixture-material-property
 KEY
        fem-mixture-composite-material-property "is made of" p
        ROLE fem-mixture-material-property-number FOR
             fem-material-property-number
        fem-mixture-volume-sequence-number
  ENDKEY
        fem-mixture-volume-fraction
        fem-mixture-volume-orientation
        fem-material-property "is referred by"
        ROLE fem-mixture-volume-material-property-number FOR
             fem-material-property-number
ENDENTITY

ENTITY  fem-halpin-tsai-material-property
 CATEGORY BY fem-composite-material-type
 OF     fem-composite-material-property
        fem-halpin-tsai-fiber-volume-fraction
        fem-halpin-tsai-matrix-volume-fraction
        fem-halpin-tsai-coefficients
        fem-material-property "is referred by"
        ROLE fem-halpin-tsai-fiber-material-property-number FOR
             fem-material-property-number
        fem-material-property "is referred by"
        ROLE fem-halpin-tsai-matrix-material-property-number FOR
             fem-material-property-number
ENDENTITY

ENTITY  fem-laminate-composite-material-property
 CATEGORY BY fem-composite-material-type
 OF     fem-composite-material-property
        fem-laminate-z-offset
ENDENTITY

ENTITY  fem-laminate-material-property
 KEY
        fem-laminate-composite-material-property "is made of" p
        fem-laminate-ply-sequence-number
 ENDKEY
        fem-material-property "is referred by"
        ROLE fem-laminate-ply-material-property-number FOR
             fem-material-property-number
        fem-laminate-ply-thickness
        fem-laminate-ply-orientation
ENDENTITY

ENTITY  fem-material-table
 CATEGORY BY fem-material-property-type
 OF     fem-material-property
        fem-material-table-independent-variable-descriptor
ENDENTITY

ENTITY  fem-material-table-instance
 KEY
        fem-material-table "has" p
        fem-material-table-independent-variable-sequence-number
 ENDKEY
        fem-material-property "has"
        ROLE fem-material-table-material-property-number FOR
             fem-material-property-number
        fem-material-table-independent-variable-value
ENDENTITY

ENDS
ENDVIEW
```

9.9 FEM EXPRESS Specification

The FEM committee of ISO TC184/SC4/WG1 have been developing their reference model using the graphical data modelling language IDEF1X. This appendix contains a translation of that reference model from IDEF1X to EXPRESS. It is used a basis for the integration of FEM into the Integrated Product Information Model.

There is no algorithm available for direct translation from IDEF1X to EXPRESS and for many of the IDEF entities there were found to be several options for modelling in express. In the cases where it was not clear which solution was the best notes are given which explain the difficulty. These can be found in section 5.

Section ?? contains a list of general rules which were followed wherever possible during the translation. They are rules which seemed to give a sensible Express model for many IDEF entities but it is not possible to apply them universally.

Also included, in section ??, is a cross reference between the IDEF1X entities and the EXPRESS entities to aid anyone comparing the two models.

9.9.1 General Rules

1. Every IDEF entity has a corresponding Express entity unless the IDEF entity exists only to resolve a many-to-many relationship between two other entities. In this case the IDEF entity will usually appear only as an attribute of one of the two other entities. An IDEF entity may also not have a corresponding Express entity under condition 6 below.
2. Where an IDEF entity has a foreign key attribute above the line and the relationship between this entity and its parent is 1 or many, then this key will not appear as an attribute of the entity in Express but the relationship is satisfied by the entity being referenced by a attribute of type set in the parent entity.
3. Where an IDEF entity has a foreign key attribute above the line and the relationship between this entity and its parent is 0, 1 or many, then this may appear either as an attribute in the child entity or as a set of entity references in the parent entity.
4. Where an IDEF entity has a foreign key attribute below the line this appears as an entity reference attribute in the corresponding Express entity.
5. Each IDEF entity, except those mentioned in 6 below, has one key attribute (above the line) which is not a foreign key. This will generally not appear in Express unless it is considered an important piece of information which it is desired to access externally.
6. Where there is a 0 or 1 relationship between entities then the child can normally be considered as an optional attribute of the parent and will appear as such in the Express model.
7. All of the information for UNITS was moved to its own model and it not in the FEM EXPRESS specification.

9.9.2 FEM Reference Model

```
SCHEMA Finite_Element_Analysis;

TYPE coordinate_pair = ARRAY [1:2] OF real;
END_TYPE;

TYPE coordinate_triple = ARRAY [1:3] OF real;
END_TYPE;

TYPE element_order = ENUMERATION OF (constant,linear,quadratic,cubic);
END_TYPE;

TYPE element_shape = ENUMERATION OF (point,
        line,
        triangle,
        quadrilateral,
        tetrahedron,
        wedge,
        pyramid,
        hexadedron);
END TYPE;

TYPE enumeration_of_constant_varying = enumeration of (constant,varying);
END_TYPE;

TYPE  coordinate_system_type = ENUMERATION OF (rh_rectangular,
                                               rh_cylindrical,
                                               rh_spherical,
                                               lh_rectangular,
                                               lh_cylindrical,
                                               lh_spherical);
END TYPE;

FUNCTION derive_maxnodes (order: element_order, shape: element_shape):integer;

(* This function is incomplete.  It gives an idea of what is required
but needs to be extended to include all possible combinations of
element_shape and element_order *)

LOCAL
  maxnodes : integer;
END_LOCAL;
```

```
    IF shape = point THEN
      maxnodes := 1;
    END_IF;
    IF (shape = line AND order < 2) THEN
      maxnodes := 2;
    END_IF;
    IF (shape = line AND order > 1) THEN
      maxnodes := order + 1;
    END_IF;
    IF (shape = triangle AND order < 2) THEN
      maxnodes := 3;
    END_IF;
    IF (shape = triangle AND order = 2) THEN
      maxnodes := 7;
    END_IF;
RETURN (maxnodes);
END_FUNCTION;

FUNCTION derive_geom_prop_list_size(attr1:enumeration_of_constant_varying,
                                                maxnodes:integer): integer;
LOCAL
  result : integer;
END_LOCAL;
  IF attr1 = constant THEN
    result := 1
  ELSE
    result := MAXNODES;
  END_IF;
RETURN (result);
END_FUNCTION;

FUNCTION test_geom_prop(attr1:enumeration_of_constant_varying;maxnodes:
maxnodes;node_list:node_list;attr4:geometric_property_ref):LOGICAL;
LOCAL
  result:LOGICAL;
END_LOCAL;
  result := TRUE;
IF attr1 = varying THEN
  REPEAT I:1 TO maxnodes WHILE result = TRUE;
  IF nodes_list(I) <> NULL THEN
    IF attr4(I) = NULL THEN
      result := FALSE;
    ENDIF;
  ENDIF;
```

```
ENDIF;
RETURN (result);
END_FUNCTION;

ENTITY finite_element_model;
  fem_id          : UNIQUE string;
  author          : INTERNAL person_name;
  creating_software       : string;
  creation_date : INTERNAL date;
  creation_time : INTERNAL time;
  descriptor      : string;
  targeted_analysis_code          : string;
  units_ref       : EXTERNAL units;
  product_item_ref        : OPTIONAL EXTERNAL SET [1:#] OF product_item;
  master_coordinate_system_ref  : EXTERNAL master_coordinate_system;
  derived_coordinate_system_ref : OPTIONAL EXTERNAL SET [1:#] OF
                                                derived_coordinate_system;
  element_ref     : EXTERNAL SET [1:#] OF element;
  node_ref        : EXTERNAL SET [1:#] OF node;
  geometric_property_ref          : EXTERNAL SET [1:#] OF geometric_property;
  material_property_ref : EXTERNAL SET [1:#] OF material_property;
  approval_ref   : OPTIONAL EXTERNAL SET  [1:#] OF approval;
  group : OPTIONAL EXTERNAL SET [1:#] OF group;
END_ENTITY;

RULE fem_identifiers FOR (finite_element_model);
  LOCAL
    coord_sys : LIST [1 : #] OF fem_coordinate_system;
  END_LOCAL;
  coord_sys = coord_sys + master_coordinate_system_ref;
  REPEAT FOR EACH derived_coordinate_system IN
        derived_coordinate_system_ref;
    coord_sys = coord_sys + derived_coordinate_system;
  END_REPEAT;
  REPEAT i := 1 TO SIZEOF(coord_sys);
    IF NOT (UNIQUE coord_sys.coordinate_system_number) THEN
       VIOLATION;
    END_IF;
  REPEAT FOR EACH element IN element_ref;
    IF NOT (UNIQUE element.element_number) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH node IN node_ref;
    IF NOT (UNIQUE node.node_number) THEN
```

```
        VIOLATION;
    END_IF;
  REPEAT FOR EACH geometric_property IN geometric_property_ref;
    IF NOT (UNIQUE geometric_property.geometric_property_id) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH material_property IN material_property_ref;
    IF NOT (UNIQUE material_property.material_property_id) THEN
        VIOLATION;
    END_IF;
  REPEAT FOR EACH group IN group_ref;
    IF NOT (UNIQUE group.group_name) THEN
        VIOLATION;
    END_IF;
  END_REPEAT;
END_RULE;

ENTITY units;
  scaled_length_unit    : INTERNAL scaled_length_unit;
  scaled_mass_unit      : INTERNAL scaled_mass_unit;
  scaled_time_unit      : INTERNAL scaled_time_unit;
  scaled_electric_current_unit  : INTERNAL scaled_current_unit;
  scaled_thermodynamic_temperature_unit : INTERNAL scaled_temperature_unit;
  scaled_amount_of_substance_unit       : INTERNAL scaled_amount_unit;
  scaled_luminous_intensity_unit        : INTERNAL scaled_luminous_intensity_unit;
  scaled_plane_angle_unit       : INTERNAL scaled_plane_angle_unit;
  scaled_solid_angle_unit       : INTERNAL scaled_solid_angle_unit;
END_ENTITY;

ENTITY date;
  year  : integer;
  month : integer;
  date  : integer;
WHERE
  year > 0;
  1 <= month <= 12;
  1 <= day <= 31;
END_ENTITY;

ENTITY time;
  hour  : INTEGER;
  minute        : INTEGER;
  second        : INTEGER;
  msec  : INTEGER;
```

```
WHERE
 0 <= hour < 24;
 0 <= minute < 60;
 0 <= second < 60;
 0 <= msec < 1000;
END_ENTITY;

ENTITY data_time;
  d      : date;
  t      : time;
END_ENTITY;

ENTITY person_name;
  name_text     : string;
END_ENTITY;

ENTITY person_and_organisation;
  person         : OPTIONAL person_name;
  company        : OPTIONAL string;
  department     : OPTIONAL string;
  section        : OPTIONAL string;
  project        : OPTIONAL string;
WHERE
   (person AND company AND department AND section AND
  project) <> NULL;
END_ENTITY;

ENTITY approval;
  approval_date : date_time;
  approving_org : person_and_organisation;
  approval_name : string;
END_ENTITY;

ENTITY element;
  element_number         : integer;
  geometric_property_coord_sys   : EXTERNAL fem_coordinate_system;
  local_coordinate_system        : EXTERNAL fem_coordinate_system;
  material_property_coord_sys    : EXTERNAL fem_coordinate_system;
  element_purpose_descriptor     : string;
  element_order_number   : element_order;
  element_shape_number   : element_shape;
  node_list      : EXTERNAL ARRAY [1 : maxnodes] OF node;
  geometric_property_type        : enumeration_of_constant_varying;
  geometric_property_ref         : EXTERNAL ARRAY [1 : geom_prop_list_size] OF
```

```
                                                               geometric_property;
WHERE
  maxnodes = derive_maxnodes (element_order,
                                         element_shape);
  geom_prop_list_size = derive_geom_prop_list_size(geometric_property_type,
                                                                  maxnodes);
  test_geom_prop(enumeration_of_constant_varying;maxnodes;node_list;
                                             geometric_property_ref) = TRUE;
END_ENTITY;

ENTITY node;
  node_number    : integer;
  coordinate_1   : real;
  coordinate_2   : real;
  coordinate_3   : real;
  definition_coordinate_system_ref       : EXTERNAL fem_coordinate_system;
  output_coordinate_system_ref   : EXTERNAL fem_coordinate_system;
END_ENTITY;

ENTITY fem_coordinate_system SUPERTYPE OF (master_coordinate_system OR
        derived_coordinate_system);
  coordinate_system_number      : integer;
  coordinate_system_type        : coordinate_system_type;
END_ENTITY;

ENTITY master_coordinate_system SUBTYPE OF (fem_coordinate_system);
END_ENTITY;

ENTITY derived_coordinate_system SUBTYPE OF (fem_coordinate_system);
  transformation_matrix : ARRAY [1:4] OF ARRAY [1:3] OF real;
  reference_coordinate_system   : EXTERNAL master_coordinate_system;
END_ENTITY;

ENTITY group;
  group_number  : integer;
  group_name    : OPTIONAL string;
  node_ref      : OPTIONAL EXTERNAL SET [1:#] OF node;
  element_ref   : OPTIONAL EXTERNAL SET [1:#] OF element;
END_ENTITY;

ENTITY geometric_property SUPERTYPE OF (point_geometric_property OR
                line_geometric_property OR
                beam_geometric_property OR
                membrane_shell_geometric_property OR
                solid_geometric_property);
  geometric_property_number     : integer;
  property_text : OPTIONAL string;
  integration_order     : OPTIONAL integer;
END_ENTITY;

ENTITY point_geometric_property SUBTYPE OF (geometric_property);
  point_mass_half_6x6_matrix    : ARRAY [1:21] OF real;
END_ENTITY;

ENTITY line_geometric_property SUBTYPE OF (geometric_property);
  spring_property_ref   : OPTIONAL EXTERNAL spring_geometric_property;
  damper_property_ref   : OPTIONAL EXTERNAL damper_geometric_property;
WHERE
  NOT((spring_property_ref  = NULL) AND (damper_property_ref  = NULL));
END_ENTITY;

ENTITY spring_geometric_property;
  property_list : EXTERNAL LIST [1:#] OF spring_property;
END_ENTITY;

ENTITY spring_property;
  dof_sequence_no_1     : integer;
  dof_sequence_no_2     : integer;
  spring_coefficient    : real;
  stress_recovery_coefficient   : real;
END_ENTITY;

ENTITY damper_geometric_property;
  property_list : EXTERNAL LIST [1:#] OF damper_property;
END_ENTITY;

ENTITY damper_property;
  dof_sequence_no_1     : integer;
  dof_sequence_no_2     : integer;
  damper_coefficient    : real;
END_ENTITY;

ENTITY beam_geometric_property SUBTYPE OF (geometric_property);
  pin_array_1_2 : OPTIONAL ARRAY [1 : 2] OF ARRAY [1 : 6] OF integer;
  offset_vector_1_2     : OPTIONAL ARRAY [1 : 2] OF coordinate_triple;
  warping_coefficients_1_2      : OPTIONAL ARRAY [1:2] OF real;
  beam_interval_ref     : EXTERNAL LIST [1 : #] OF beam_interval;
END_ENTITY;
```

```
ENTITY beam_interval;
  beam_length_fraction_B        : real;
  section_ref_A : EXTERNAL beam_section;
  section_ref_B : EXTERNAL beam_section;
  material_ref  : EXTERNAL material_property;
WHERE
  beam_length_fraction_B BETWEEN 0. AND 1.;
END_ENTITY;

ENTITY beam_section SUPERTYPE OF (beam_property_data OR
        standard_beam_section);
  stress_recovery_coefficients  : OPTIONAL LIST [1:#] OF coordinate_pair;
END_ENTITY;

ENTITY beam_property_data SUBTYPE OF (beam_section);
  cross_sectional_area  : real;
  second_moment_of_area_I11     : real;
  second_moment_of_area_I22     : real;
  second_moment_of_area_I12     : real;
  polar_moment  : real;
  shear_area_K1 : real;
  shear_area_K2 : real;
  shear_relief_coefficient_S1   : OPTIONAL real;
  shear_relief_coefficient_S2   : OPTIONAL real;
  non_structural_mass   : OPTIONAL real;
  centroid_offset       : OPTIONAL coordinate_pair;
  shear_centre_offset   : OPTIONAL coordinate_pair;
  non_structural_mass_offset    : OPTIONAL coordinate_pair;
END_ENTITY;

ENTITY standard_beam_section SUBTYPE OF (beam_section);
  property_data : external_reference;
END_ENTITY;

ENTITY membrane_shell_geometric_property SUBTYPE OF (geometric_property);
  membrane_material     : EXTERNAL material_property;
  thickness     : real;
  nonstructural_mass    : real;
  bending_material      : OPTIONAL EXTERNAL material_property;
  shear_material        : OPTIONAL EXTERNAL material_property;
  membrane-bending_material     : OPTIONAL EXTERNAL material_property;
  second_moment_of_area : OPTIONAL real;
  shear_thickness       : OPTIONAL real;
  z_offset        : OPTIONAL real;
  stress_recovery_coefficient   : OPTIONAL SET OF real;
END_ENTITY;

ENTITY solid_geometric_property SUBTYPE OF (geometric_property);
  material        : EXTERNAL material_property;
END_ENTITY;

ENTITY material_property SUPERTYPE OF (homogeneous_material OR
                composite_material OR
                 material_table);
  material_property_number      : integer;
  text  : OPTIONAL string;
END_ENTITY;

ENTITY composite_material SUPERTYPE OF (halpin-tsai_material OR
                laminate_composite_material OR
                mixture_composite_material)
        SUBTYPE OF (material_property);
END_ENTITY;

ENTITY mixture_composite_material SUBTYPE OF (composite_material);
  mixture_material       : EXTERNAL LIST [1:#] OF mixture_material_property;
WHERE
  sum(mixture_material.mixture_fraction) = 1.0;
END_ENTITY;

ENTITY mixture_material_property;
  volume_fraction        : real;
  volume_orientation     : real;
  volume_material        : EXTERNAL material_property;
WHERE;
  0. < volume_fraction < 1.;
END_ENTITY;

ENTITY halpin-tsai-material SUBTYPE OF (composite_material);
  fiber_volume_fraction : real;
  matrix_volume_fraction        : real;
  coefficients  : LIST [1:#] OF real;
  fiber_material         : EXTERNAL material_property;
  matrix_material        : EXTERNAL material_property;
WHERE
  fiber_volume_fraction + matrix_volume_fraction = 1.0;
END_ENTITY;
```

```
ENTITY laminate_composite_material SUBTYPE OF (composite_material);
  laminate_z_offset       : real;
  laminate_material       : EXTERNAL LIST [1:#] OF laminate_material_property;
END_ENTITY;

ENTITY laminate_material_property;
  ply_thickness : real;
  ply_orientation        : real;
  ply_material   : EXTERNAL material_property;
END ENTITY;

ENTITY material_table SUBTYPE OF (material_property);
  independent_variable_descriptor        : string;
  material_table_instance_ref    : EXTERNAL LIST [1:#] OF material_table_instance;
END_ENTITY;

ENTITY material_table_instance;
  material_table_material_ref    : EXTERNAL material_property;
  independent_variable_value     : real;
END_ENTITY;

ENTITY homogeneous_material SUPERTYPE OF (isotropic_material OR
                  orthotropic_material OR
                  anisotropic_material);
        SUBTYPE OF (material_property);
  mass_density   : OPTIONAL real;
  structural_damping_coefficient         : OPTIONAL real;
END_ENTITY;

ENTITY isotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property        : OPTIONAL EXTERNAL isotropic_thermal_property;
  structural_property     : OPTIONAL EXTERNAL isotropic_structural_property;
  expansion_property      : OPTIONAL EXTERNAL isotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)  AND
  (expansion_property = NULL));
END_ENTITY;

ENTITY isotropic_thermal_property;
  thermal_conductivity_coefficient       : real;
  specific_heat_capacity         : real;
  convective_film_coefficient    : real;
END_ENTITY;
```

```
ENTITY isotropic_structural_property;
  Youngs_modulus        : real;
  poisson's_ratio       : real;
  shear_modulus : real;
END_ENTITY;

ENTITY isotropic_thermal_expansion;
  thermal_expansion_coefficient : real;
  thermal_expansion_reference_temp      : real;
END_ENTITY;

ENTITY orthotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property      : OPTIONAL EXTERNAL orthotropic_thermal_property;
  structural_proerty    : OPTIONAL EXTERNAL orthotropic_structural_property;
  expansion_property    : OPTIONAL EXTERNAL orthotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)  AND
  (expansion_property = NULL));
END_ENTITY;

ENTITY orthotropic_structural_property SUPERTYPE OF
                                             (orthotropic_structural_2d_property OR
                                              orthotropic_structural_3d_property);
  in-plane_youngs_moduli        : ARRAY [1:2] OF real;
  in-plane_poissons_ratio       : real;
  in-plane_shear_modulus        : real;
END_ENTITY;

ENTITY orthotropic_structural_2d_property SUBTYPE OF
                                             (orthotropic_STRUCTURAL_property);
END_ENTITY;

ENTITY orthotropic_structural_3d_property SUBTYPE OF
                                             (orthotropic_structural_property);
  out-of-plane_youngs_moduli    : ARRAY [1:2] OF real;
  out-of-plane_poissons_ratio   : ARRAY [1:2] OF real;
  out-of-plane_shear_modulus    : real;
END_ENTITY;

ENTITY orthotropic_thermal_property SUPERTYPE OF
                                               (orthotropic_thermal_2d_property OR
                                                orthotropic_thermal_3d_property);
  in-plane_thermal_conductivities       : ARRAY [1:2] OF real;
  specific_heat_capacity        : real;
  convective_film_coefficient   : real;
END_ENTITY;

ENTITY orthotropic_thermal_2d_property SUBTYPE OF
                                         (orthotropic_thermal_property);
END_ENTITY;

ENTITY orthotropic_thermal_3d_property SUBTYPE OF
                                         (orthotropic_thermal_property);
  out-of-plane_thermal_conductivity     : real;
END_ENTITY;

ENTITY orthotropic_thermal_expansion SUPERTYPE OF
                                     (orthotropic_2d_thermal_expansion OR
                                      orthotropic_3d_thermal_expansion);
  in-plane_thermal_expansion_coefficients       : ARRAY [1:2] OF real;
  thermal_expansion_reference_temperature       : real;
END_ENTITY;

ENTITY orthotropic_2d_thermal_expansion SUBTYPE OF
                                     (orthotropic_thermal_expansion);
END_ENTITY;

ENTITY orthotropic_3d_thermal_expansion SUBTYPE OF
                                     (orthotropic_thermal_expansion);
  out-of-plane_thermal_expansion_coefficient    : real;
END_ENTITY;

ENTITY anisotropic_material SUBTYPE OF (homogeneous_material);
  thermal_property      : OPTIONAL EXTERNAL anisotropic_thermal_property;
  structural_proerty    : OPTIONAL EXTERNAL anisotropic_structural_property;
  expansion_property    : OPTIONAL EXTERNAL anisotropic_thermal_expansion;
WHERE
  NOT (( thermal_property = NULL) AND (structural_property = NULL)  AND
  (expansion_property = NULL));
END_ENTITY;

ENTITY anisotropic_structural_property SUPERTYPE OF
                                    (anisotropic_structural_2d_property OR
                                     anisotropic_structural_3d_property);
END_ENTITY;

ENTITY anisotropic_structural_2d_property SUBTYPE OF
```

```
                                         (anisotropic_structural_property);
  material_elasticity_half_3x3_matrix    : ARRAY [1:6] OF real;
END_ENTITY;

ENTITY anisotropic_structural_3d_property SUBTYPE OF
                                         (anisotropic_structural_property);
  material_elasticity_half_6x6_matrix    : ARRAY [1:21] OF real;
END_ENTITY;

ENTITY anisotropic_thermal_property SUPERTYPE OF
                                         (anisotropic_2d_thermal_property OR
                                          anisotropic_thermal_3d_property);
  specific_heat_capacity        : real;
END_ENTITY;

ENTITY anisotropic_thermal_2d_property SUBTYPE OF
                                         (anisotropic_thermal_property);
  thermal_conductivity  : ARRAY [1 TO 2] OF real;
END_ENTITY;

ENTITY anisotropic_thermal_3d_property SUBTYPE OF
                                         (anisotropic_thermal_property);
  thermal_conductivity  : ARRAY [1:3] OF real;
END_ENTITY;

ENTITY anisotropic_thermal_expansion SUPERTYPE OF
                                         (anisotropic_2d_thermal_expansion OR
                                          anisotropic_3d_thermal_expansion);
  thermal_expansion_reference_temp      : real;
END_ENTITY;

ENTITY anisotropic_2d_thermal_expansion SUBTYPE OF
                                         (anisotropic_thermal_expansion);
  thermal_expansion_coefficients        : ARRAY [1:3] OF real;
END_ENTITY;

ENTITY.anisotropic_3d_thermal_expansion SUBTYPE OF
                                         (anisotropic_thermal_expansion);
  thermal_expansion_coefficients        : ARRAY [1:6] OF real;
END_ENTITY;

END_SCHEMA;
```

9.9.3 IDEF/EXPRESS Entity Cross Reference List

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM ENVIRONMENT			1
FEA RESULTS			1
FEM	finite_element_model		
UNITS	units		
PRODUCT ITEM/FEM	finite_element_model	product_item_ref	
FEA CONTROL	fea_control		
FEA	fea		
FEM APPROVAL	approval		3
FEA CONTROL APPROVAL	approval		3
FEM ELEMENT	element		
FEM NODE	node		
FEM CONNECTIVITY	element	node_list	
FEM COORDINATE SYSTEM	fem coordinate_system		
FEM COORDINATE MASTER SYSTEM	master_coordinate_system		
FEM COORDINATE DEFINITION SYSTEM	derived_coordinate_system		
FEM COORDINATE TRANSFORM	derived_coordinate_system	transformation_matrix	

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM GROUP	group		
FEM NODE GROUP	group	node_ref	
FEM ELEMENT GROUP	group	element_ref	
FEM GEOMETRIC PROPERTY	geometric_property		
FEM ELEMENT GEOMETRIC PROPERTY	element	geometric_property_ref	2
FEM ELEMENT/NODE GEOMETRIC PROPERTY	element	geometric_property_ref	2
FEM GEOMETRIC PROPERTY TEXT	geometric_property	property_text	
FEM INTEGRATION ORDER	geometric_property	integration_order	
FEM POINT GEOMETRIC PROPERTY	point_geometric_property		
FEM LINE ELEMENT GEOMETRIC PROPERTY	line_geometric_property		
FEM SPRING GEOMETRIC PROPERTY	spring_geometric_property		
FEM SPRING PROPERTY	spring_property		
FEM DAMPER GEOMETRIC PROPERTY	damper_geometric_property		
FEM DAMPER PROPERTY	damper_property		
FEM BEAM GEOMETRIC PROPERTY	beam_geometric_property		

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM BEAM WARPING DATA	beam_geometric_property	warping_coefficients_1_2	
FEM BEAM OFFSET VECTORS	beam_geometric_property	offset_vector_1_2	
FEM BEAM PIN DATA	beam_geometric_property	pin_array_1_2	
FEM BEAM INTERVAL	beam_interval		
FEM BEAM SECTION	beam_section		
FEM BEAM STRESS RECOVERY COEFFICIENT	beam_section	stress_recovery_coefficients	
FEM BEAM PROPERTY DATA	beam_property_data		
FEM BEAM STANDARD SECTION	standard beam_section		
FEM MEMBRANE/SHELL GEOMETRIC PROPERTY		membrane_shell_geometric_property	
FEM SHELL GEOMETRIC PROPERTY	membrane_shell_geometric_property	bending_material shear_material membrane-bending_material second_moment_of_area shear_thickness z_offset	
FEM SHELL STRESS RECOVERY COEFFICIENT	membrane_shell_geometric_property	stress_recovery_coefficient	
FEM SOLID GEOMETRIC PROPERTY	solid_geometric_property		
FEM MATERIAL PROPERTY	material_property		

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM COMPOSITE MATERIAL PROPERTY	composite_material		
FEM MIXTURE COMPOSITE MATERIAL PROPERTY	mixture_composite_material		
FEM MIXTURE MATERIAL PROPERTY	mixture_material_property		
FEM HALPIN-TSAI MATERIAL PROPERTY	halpin-tsai_material		
FEM LAMINATE COMPOSITE MATERIAL PROPERTY	laminate_composite_material		
FEM LAMINATE MATERIAL PROPERTY	laminate_material_property		
FEM MATERIAL TABLE	material_table		
FEM MATERIAL TABLE INSTANCE		material_table_instance	
FEM HOMOGENEOUS MATERIAL PROPERTY	homogeneous_material		
FEM STRUCTURAL MASS DENSITY	homogeneous_material	mass_density	
FEM STRUCTURAL DAMPING COEFFICIENT	homogeneous_material	structural_damping_coefficient	
FEM ISOTROPIC MATERIAL PROPERTY	isotropic_material		
FEM ISOTROPIC THERMAL MATERIAL PROPERTY	isotropic_thermal_property		

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM ISOTROPIC STRUCTURAL MATERIAL PROPERTY	isotropic_structural_property		
FEM ISOTROPIC THERMAL EXPANSION	isotropic_thermal_expansion		
FEM ORTHOTROPIC MATERIAL PROPERTY	orthotropic_material		
FEM ORTHOTROPIC STRUCTURAL MATERIAL PROPERTY	orthotropic_structural_property		
FEM ORTHOTROPIC STRUCTURAL 2D MATERIAL PROPERTY	orthotropic_structural_2d_property		
FEM ORTHOTROPIC STRUCTURAL 3D MATERIAL PROPERTY	orthotropic_structural_3d_property		
FEM ORTHOTROPIC THERMAL MATERIAL PROPERTY	orthotropic_thermal_property		
FEM ORTHOTROPIC THERMAL 2D MATERIAL PROPERTY	orthotropic_thermal_2d_property		
FEM ORTHOTROPIC THERMAL 3D MATERIAL PROPERTY	orthotropic_thermal_3d_property		
FEM ORTHOTROPIC THERMAL EXPANSION	orthotropic_thermal_expansion		
FEM ORTHOTROPIC 2D THERMAL EXPANSION	orthotropic_2d_thermal_expansion		
FEM ORTHOTROPIC 3D THERMAL EXPANSION	orthotropic_3d_thermal_expansion		

IDEF Entity	**EXPRESS Entity**	**EXPRESS Attribute**	**Note**
FEM ANISOTROPIC MATERIAL PROPERTY	anisotropic_material		
FEM ANISOTROPIC STRUCTURAL MATERIAL PROPERTY	anisotropic_structural_property		
FEM ANISOTROPIC STRUCTURAL 2D MATERIAL PROPERTY	anisotropic_2d_structural_property		
FEM ANISOTROPIC STRUCTURAL 3D MATERIAL PROPERTY	anisotropic_3d_structural_property		
FEM ANISOTROPIC THERMAL MATERIAL PROPERTY	anisotropic_thermal_property		
FEM ANISOTROPIC THERMAL 2D MATERIAL PROPERTY	anisotropic_thermal_2d_property		
FEM ANISOTROPIC THERMAL 3D MATERIAL PROPERTY	anisotropic_thermal_3d_property		
FEM ANISOTROPIC THERMAL EXPANSION	anisotropic_thermal_expansion		
FEM ANISOTROPIC 2D THERMAL EXPANSION	anisotropic_2d_thermal_expansion		
FEM ANISOTROPIC 3D THERMAL EXPANSION	anisotropic_3d_thermal_expansion		
FEM ELEMENT/VERTEX	element_vertex		
FEM ELEMENT/EDGE	element_edge		
FEM ELEMENT/FACE	element_face		

IDEF Entity	EXPRESS Entity	EXPRESS Attribute	Note
FEM ELEMENT/SHELL	element_shell		
FEM NODE/VERTEX	node_vertex		
FEM NODE/EDGE	node_edge		
FEM NODE/FACE	node_face		
FEM NODE/SHELL	node_shell		

Notes

1. These entities are shown on the FEM planning model but are not attributed yet, therefore they have not been included in the Express model.
2. The two IDEF entities FEM ELEMENT GEOMETRIC PROPERTY and FEM ELEMENT/NODE GEOMETRIC PROPERTY were only needed to resolve the two possibilities of having either one geometric property assigned over the whole element or different geometric properties assigned to each node of the element. In the Express model this relationship is defined within the entity element. An extra attribute is required called geometric property type followed by an ARRAY of references to geometric properties the size of which is either 1 or maxnodes.
3. The two IDEF entities FEM APPROVAL and FEA CONTROL APPROVAL had the same attributes so they have been translated into one Express entity, approval.

This model is not available at this time.

N288

SECTION 9: FEM INFORMATION MODEL

10 UNITS INFORMATION MODEL

10.1 Scope & Purpose

This document defines and describes the IDEF1X reference model for units of measure used throughout data models. In addition to the IDEF1X data model there are two corresponding models. One is in EXPRESS (Section 10.10 for the complete EXPRESS listing) and the other is in Structural Modeling Language (SML) (Section 10.9 for the complete SML listing).

10.1.1 Purpose

The purpose of this document is to propose an IDEF1X reference model, together with the matching EXPRESS and SML models for units of measure which is to be submitted to the ISO TC 184/SC4 for inclusion in this international standard.

10.1.2 Scope and Viewpoint

The scope of this document is limited to the initial work done by the Finite Element Model (FEM) committee on fundamental units of measure and the input received from the other committees.

Inclusion of other units of measure information is left to a future version of this specification if needed by other application committees.

The viewpoint is that of the (FEM) committee creating this document with the broader needs for units of measure.

10.1.3 Fundamental Concepts and Assumptions

The UNITS model fundamental concepts and assumptions are presented in this section in the form of rules. The following is a table of contents for these ideas.

Concept	Rule Description
UNITS/R-1	UNITS Identification
UNITS/R-2	Fundamental Units of Measure
UNITS/R-3	UNITS Entity
UNITS/R-4	Scale Factors for Units
UNITS/R-5	Standard Sets of Fundamental Units of Measure

UNITS/R-1 UNITS Identification

Any change to a UNITS (e.g., changes to length units, mass units, scale factors, etc.) causes the occurrence of a new unit entity that must be issued a new UNITS ID.

Date Created: 24th Aug 1988 by FEM Committee

SECTION 10: UNITS INFORMATION MODEL

UNITS/R-2 Fundamental Units of Measure

There are seven base or fundamental units of measure and two supplementary units of measure that exist from which all other units of measure are derived.

The fundamental units of measure are length, mass, time, electric current, thermodynamic temperature, amount of substance and luminous intensity. The two supplementary units are plane angle and solid angle.

Date Created: 16th July 1986 by FEM Committee
Related Issue: UNITS/Issue-1 Fundamental Units of Measure

UNITS/R-3 UNITS entity

The UNITS (UNIT-1) entity is used whenever it is desired to impose the consistent use of fundamental units of measure throughout a model, such as FEM. Therefore all of the fundamental units of measure and the two supplementary units of measure are combined into this one entity.

Date Created: 16th July 1986 by FEM Committee

UNITS/R-4 Scale Factors for Units

Each fundamental unit of measure has a multiplicative scale factor to allow for the choice by a person to choose a nonstandard unit of measure by selecting a standard unit of measure and scaling it. For example, a FEM analyst wants a unit of length to be half a foot, he would select a unit of length of a foot and a scale factor of 0,5.

Date Created: 15th July 1988 by FEM Committee

UNITS/R-5 Standard Sets of Fundamental Units of Measure

There are at least two sets of standard fundamental units of measure and they are:

Units Id	**Description**
SI	ISO set of fundamental units of measure
British	British set of fundamental units of measure

See the UNITS entity in section 3 for the values of the fundamental units of measure in these sets.

Date Created: 24th Aug 1988 by FEM Committee

10.1.4 Key Abbreviations and Acronyms

FEM	Finite Element Model
ISO	International Standard Organization
PDES	Product Data Exchange Specification
SI	System Internationale
SML	Structural Modeling Language

10.2 The UNITS Planning Model

This section is intentionally left blank because it was not needed for this data model.

10.3 UNITS Reference Data Models

The UNITS entity pool and reference models used to create the IDEF1X UNITS global model are found in this section. Each reference model investigates a subset of units of measure information.

Section 10.3.1 contains a list of the data entities used to create the UNITS reference models. The actual models are presented in Section 10.3.2. The entity glossary follow in Section 10.3.3.

10.3.1 UNITS Reference Model Entity Pool

The following are the entities used in the UNITS reference and global data models.

Entity	Entity Name (Sorted by entity number)
UNIT-1	UNITS
UNIT-2	SCALED LENGTH UNIT
UNIT-3	SCALED MASS UNIT
UNIT-4	SCALED TIME UNIT
UNIT-5	SCALED ELECTRIC CURRENT UNIT
UNIT-6	SCALED THERMODYNAMIC TEMPERATURE UNIT
UNIT-7	SCALED AMOUNT OF SUBSTANCE UNIT
UNIT-8	SCALED LUMINOUS INTENSITY UNIT
UNIT-9	SCALED PLANE ANGLE UNIT
UNIT-10	SCALED SOLID ANGLE UNIT

Entity	**Entity Name (Sorted by entity name)**
UNIT-7	SCALED AMOUNT OF SUBSTANCE UNIT
UNIT-5	SCALED ELECTRIC CURRENT UNIT
UNIT-2	SCALED LENGTH UNIT
UNIT-8	SCALED LUMINOUS INTENSITY UNIT
UNIT-3	SCALED MASS UNIT
UNIT-9	SCALED PLANE ANGLE UNIT
UNIT-10	SCALED SOLID ANGLE UNIT
UNIT-6	SCALED THERMODYNAMIC TEMPERATURE UNIT
UNIT-4	SCALED TIME UNIT
UNIT-1	UNITS

10.3.2 UNITS Reference Models

This section contains all of the individual UNITS reference models used to create the UNITS global IDEF1X data model. The following list tabulates the name and figure number of each UNITS reference model.

Figure Number	Reference Model Name
150	Set of fundamental units of measure
151	Unit of length
152	Unit of mass
153	Unit of time
154	Unit of current
155	Unit of temperature
156	Unit of amount of substance
157	Unit of luminous intensity
158	Unit of plane angle
159	Unit of solid angle

SET OF_FUNDAMENTAL_UNITS_OF_MEASURE

Figure 150 depicts what the UNITS entity contains in the way of fundamental units of measure. Units rules UNITS/R-1, 2, and 3 apply to this IDEF1X diagram.

FUNDAMENTAL UNITS OF MEASURES

The next nine figures present the nine fundamental units of measure. These are:

1. UNIT OF LENGTH, Figure 151
2. UNIT OF MASS, Figure 152
3. UNIT OF TIME, Figure 153
4. UNIT OF ELECTRIC CURRENT, Figure 154
5. UNIT OF THERMODYNAMIC TEMPERATURE, Figure 155
6. UNIT OF AMOUNT OF SUBSTANCE, Figure 156
7. UNIT OF LUMINOUS INTENSITY, Figure 157
8. UNIT OF PLANE ANGLE, Figure 158
9. UNIT OF SOLID ANGLE, Figure 159

UNITS rules UNITS/R-1, 2, 3 and 4 apply to these IDEF1X diagrams.

UNITS / UNIT-1

UNITS ID

LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-150: Set of Fundamental Units of Measure

10.3.3 Entity Glossary and Business Rules

The following sections describe each data entity in the UNITS reference model. These sections are presented in numerical order by entity number.

(For detail definitions of the unit of measure shown in this section will be found in section 10.8

Entity Name: **UNITS**

Entity Number: **UNIT-1**

The UNITS data entity defines the base (or fundamental) units of measure that are to be use for deriving the other units of measure. For example, a finite element model would have one occurrence of this entity where all the data in the model would be expressed in of the sets of nine fundamental units of measure (length, mass, time, electric current, thermodynamic temperature, amount of substance, luminous intensity, plane angle and solid angle) defined by the units entity.

There are two sets of standard fundamental units of measure. They are as follows (with the corresponding values):

Standard set of fundamental units of measure - SI

Attributes	Values
LENGTH UNIT SCALE FACTOR	1.0
BASE LENGTH UNIT	Meter
MASS UNIT SCALE FACTOR	1.0
BASE MASS UNIT	Kilogram
TIME UNIT SCALE FACTOR	1.0
BASE TIME UNIT	Second
ELECTRIC CURRENT UNIT SCALE FACTOR	1.0
BASE ELECTRIC CURRENT UNIT	Ampere
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR	1.0
BASE THERMODYNAMIC TEMPERATURE UNIT	Kelvin
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR	1.0
BASE AMOUNT OF SUBSTANCE UNIT	Mole
LUMINOUS INTENSITY UNIT SCALE FACTOR	1.0
BASE LUMINOUS INTENSITY UNIT	Candela
PLANE ANGLE UNIT SCALE FACTOR	1.0
BASE PLANE ANGLE UNIT	Radian
SOLID ANGLE UNIT SCALE FACTOR	1.0
BASE SOLID ANGLE UNIT	Steradian

N288

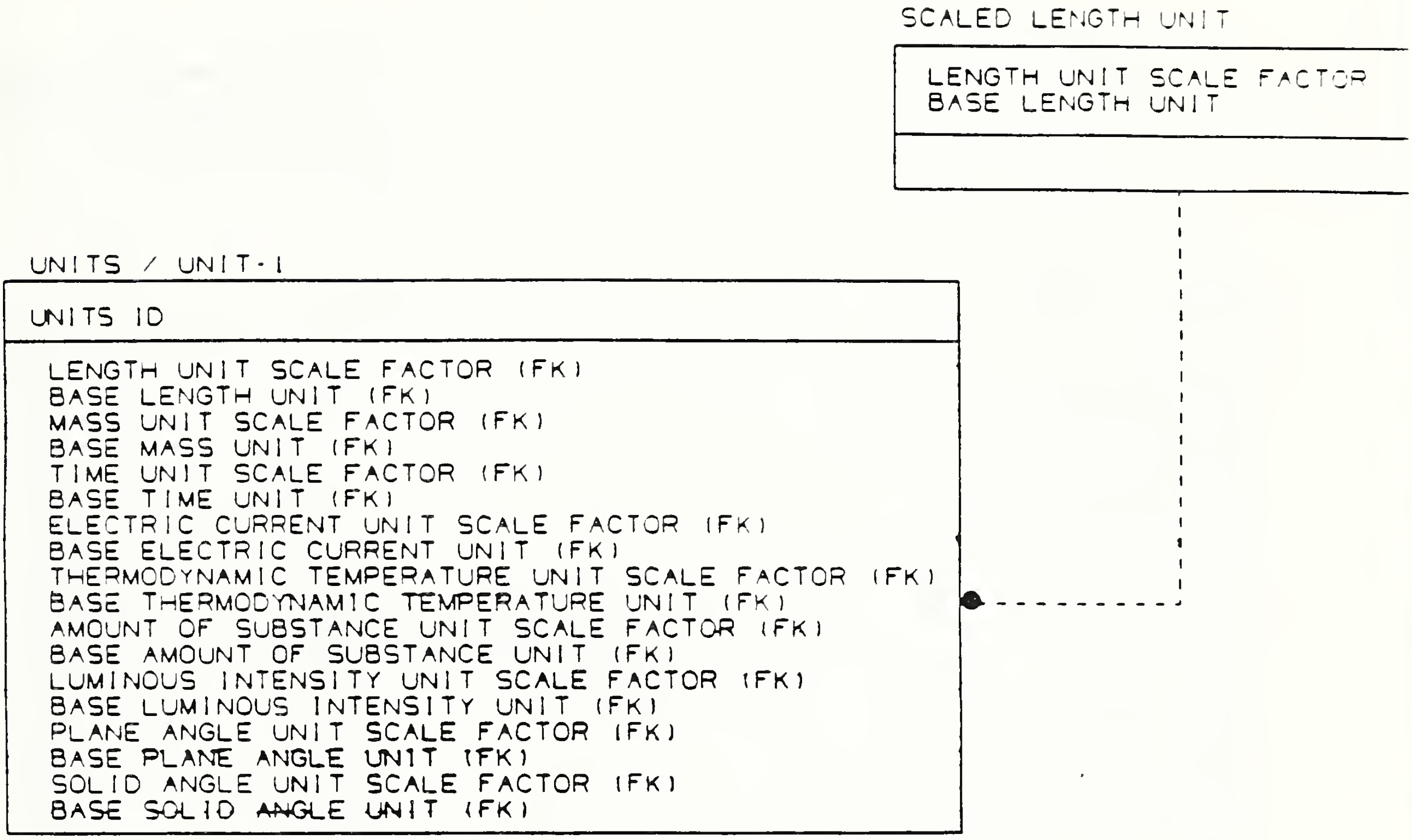

SCALED LENGTH UNIT
LENGTH UNIT SCALE FACTOR
BASE LENGTH UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-151: Unit of Length

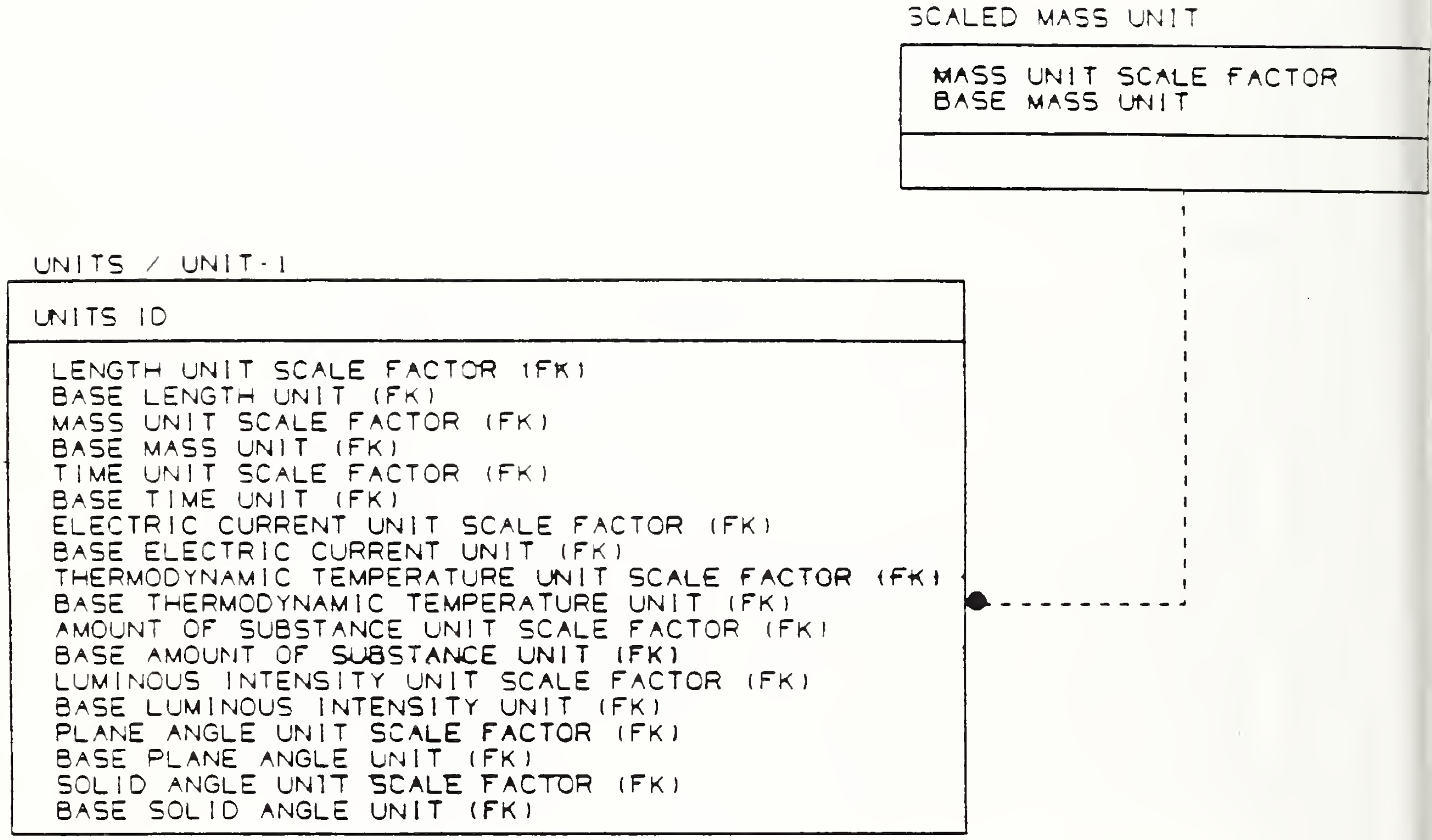

SCALED MASS UNIT
MASS UNIT SCALE FACTOR
BASE MASS UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-152: Unit of Mass

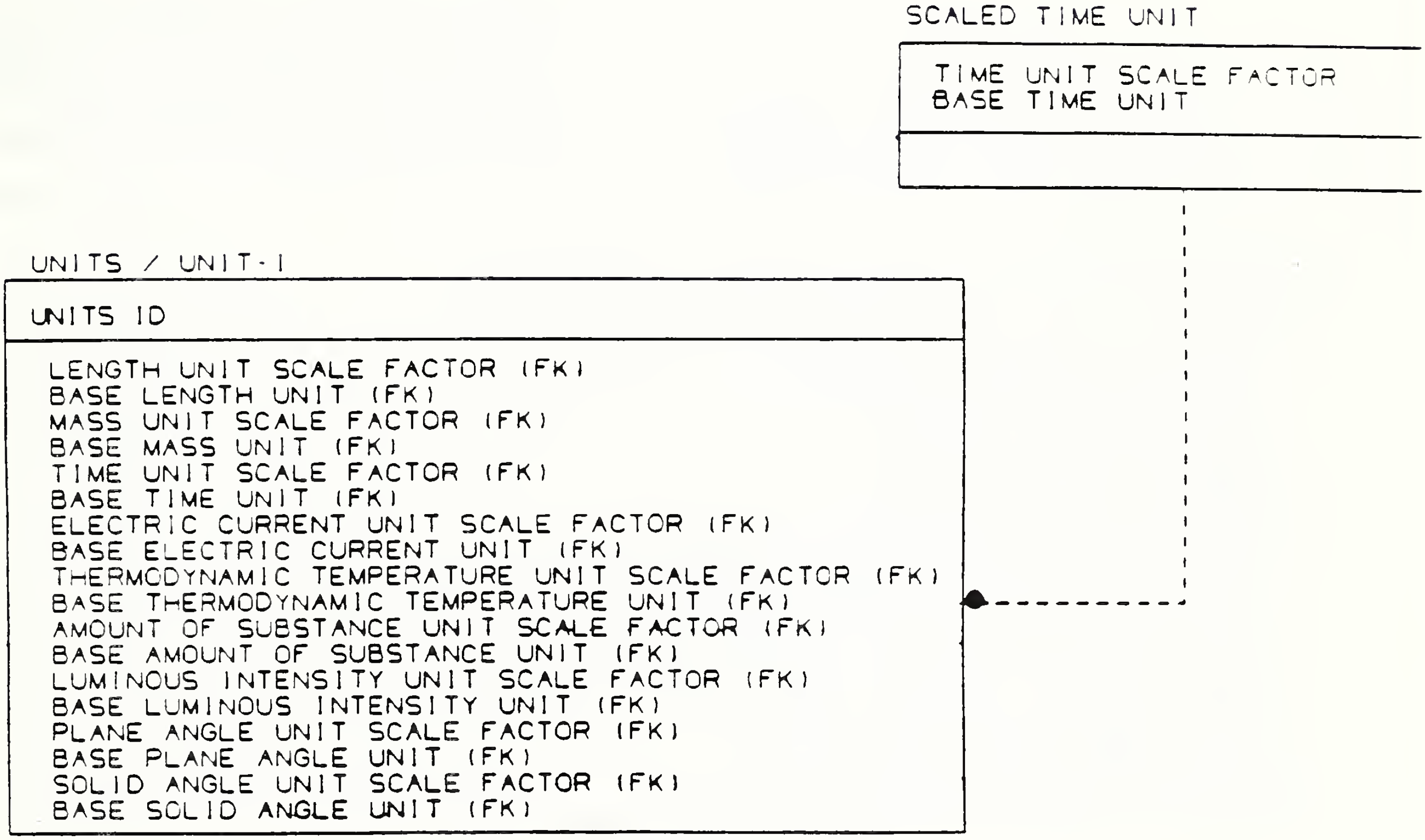

SCALED TIME UNIT
TIME UNIT SCALE FACTOR
BASE TIME UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-153: Unit of Time

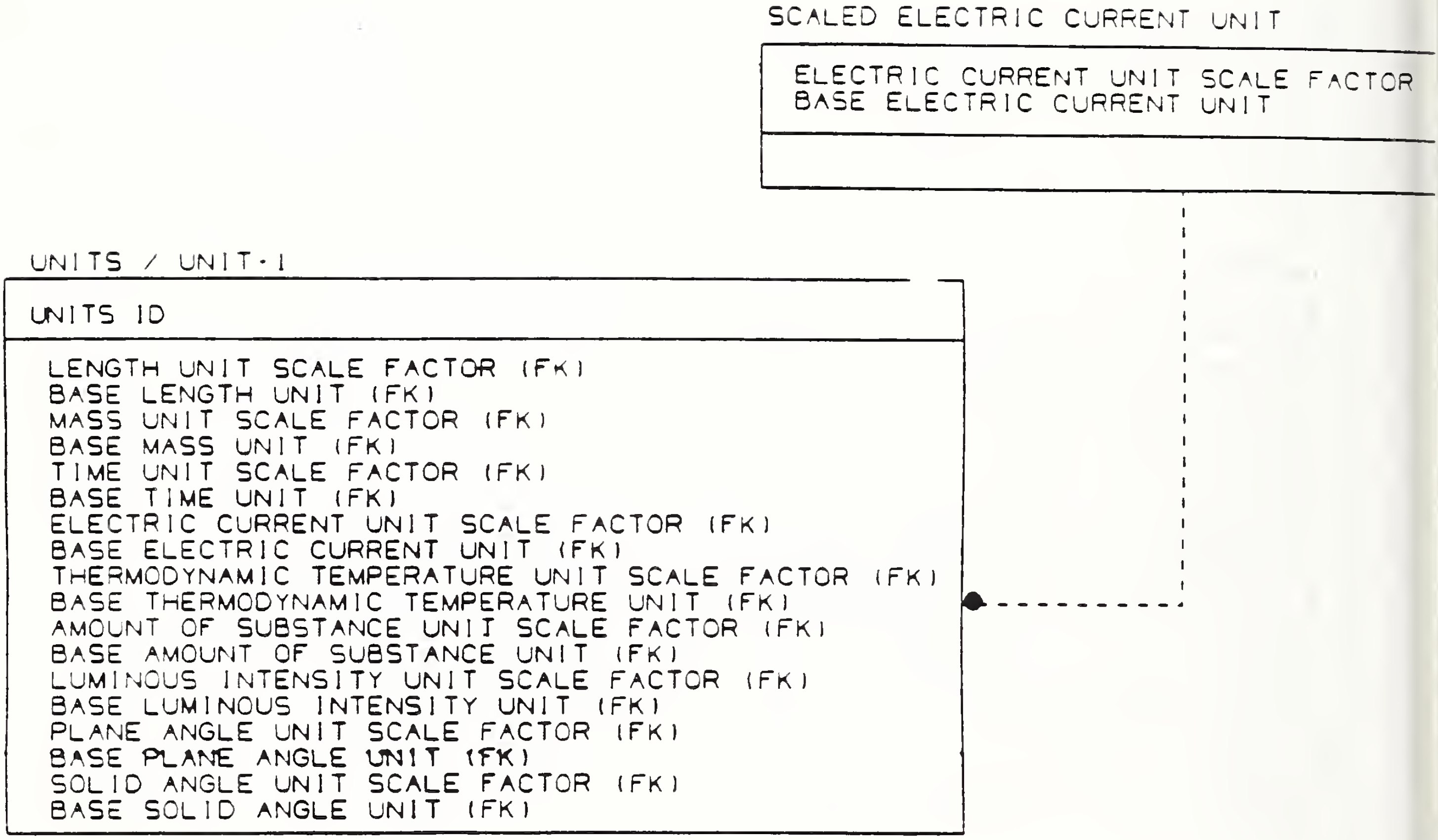

SCALED ELECTRIC CURRENT UNIT
ELECTRIC CURRENT UNIT SCALE FACTOR
BASE ELECTRIC CURRENT UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-154: Unit of Electric Current

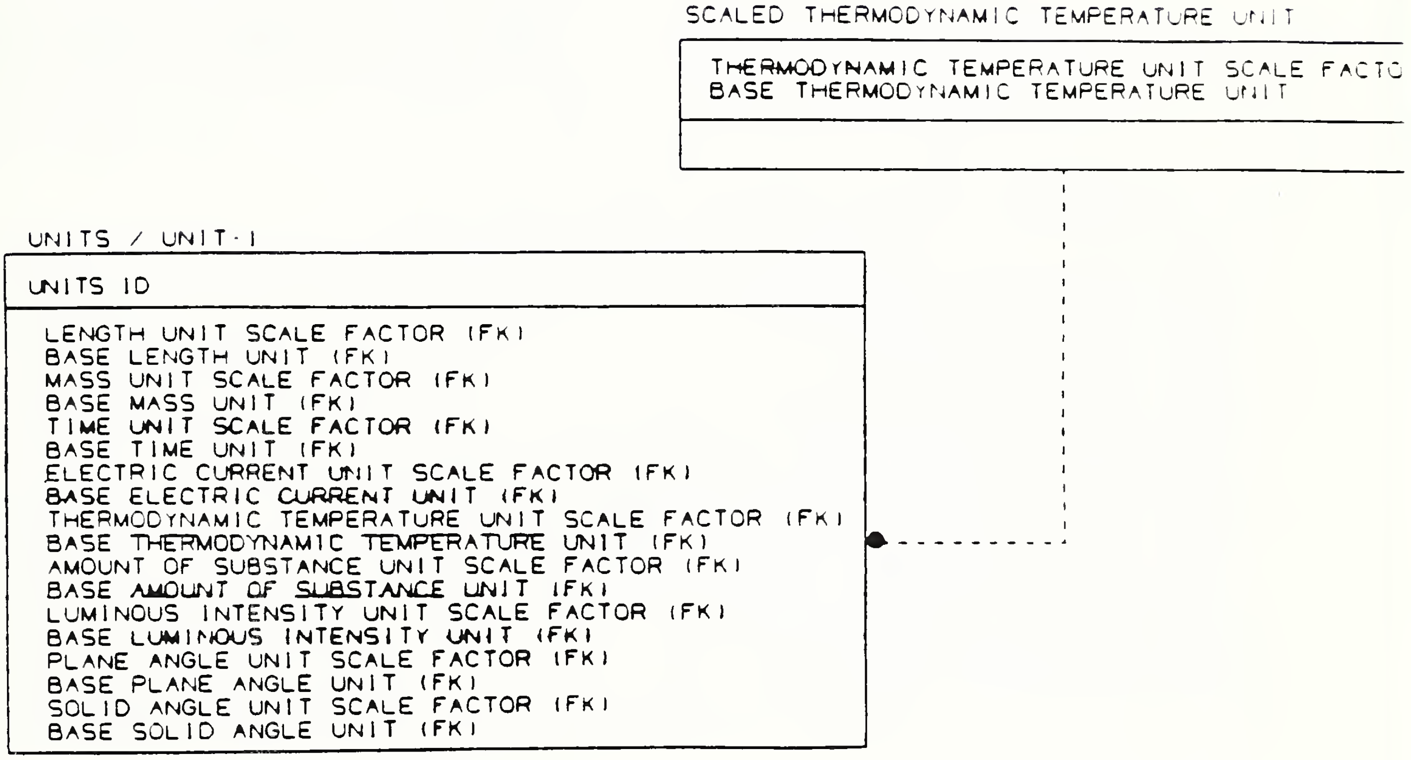

SCALED THERMODYNAMIC TEMPERATURE UNIT
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTO
BASE THERMODYNAMIC TEMPERATURE UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-155: Unit of Thermodynamic Temperature

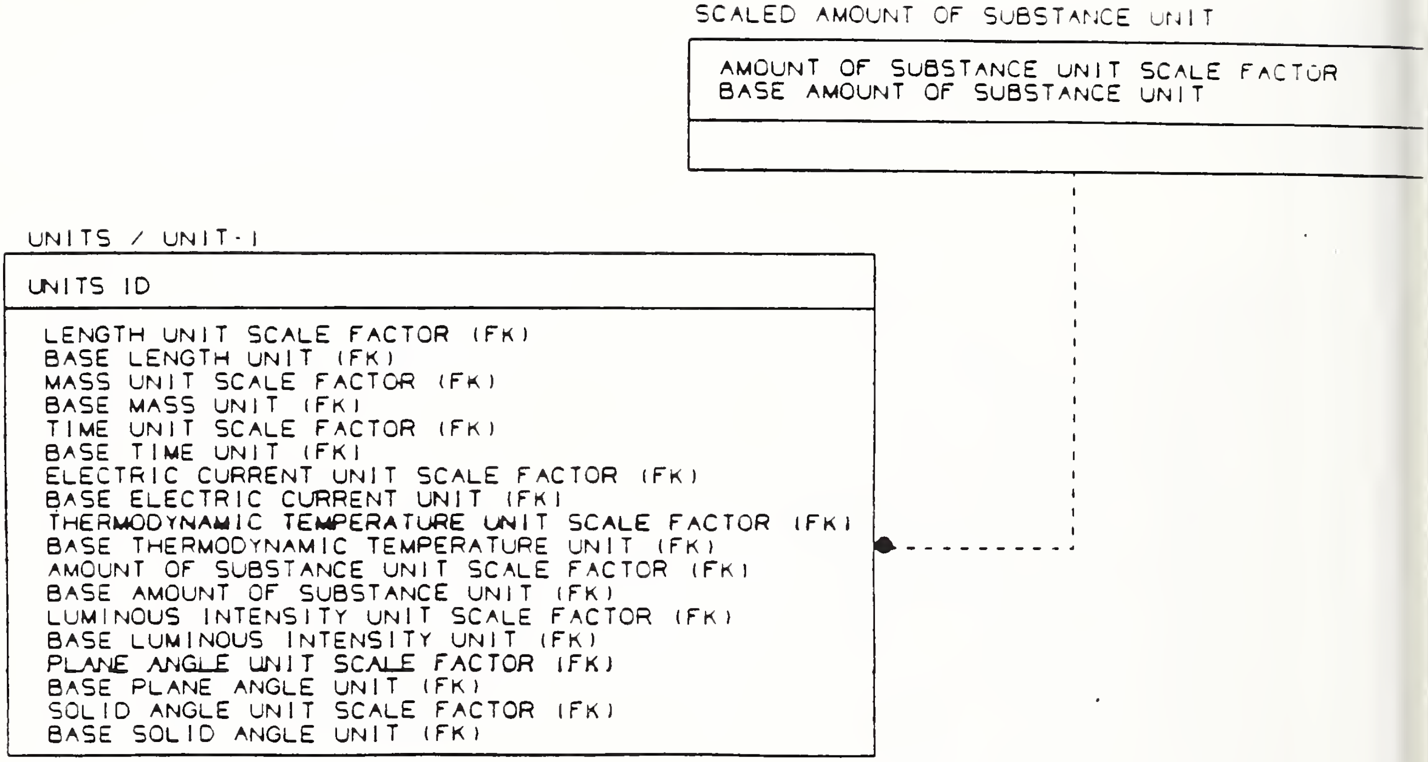

SCALED AMOUNT OF SUBSTANCE UNIT
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR
BASE AMOUNT OF SUBSTANCE UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-156: Unit of Amount of Substance

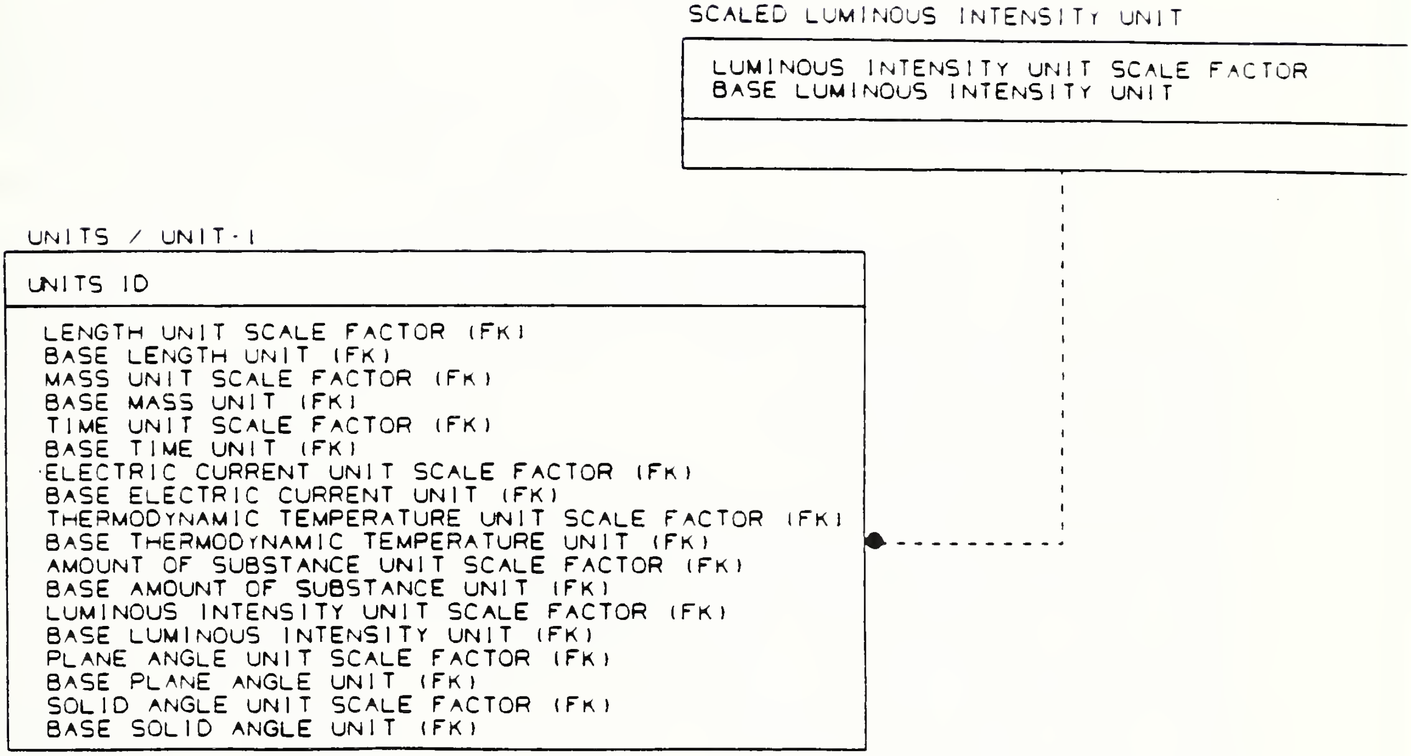

SCALED LUMINOUS INTENSITY UNIT
LUMINOUS INTENSITY UNIT SCALE FACTOR
BASE LUMINOUS INTENSITY UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-157: Unit of Luminous Intensity

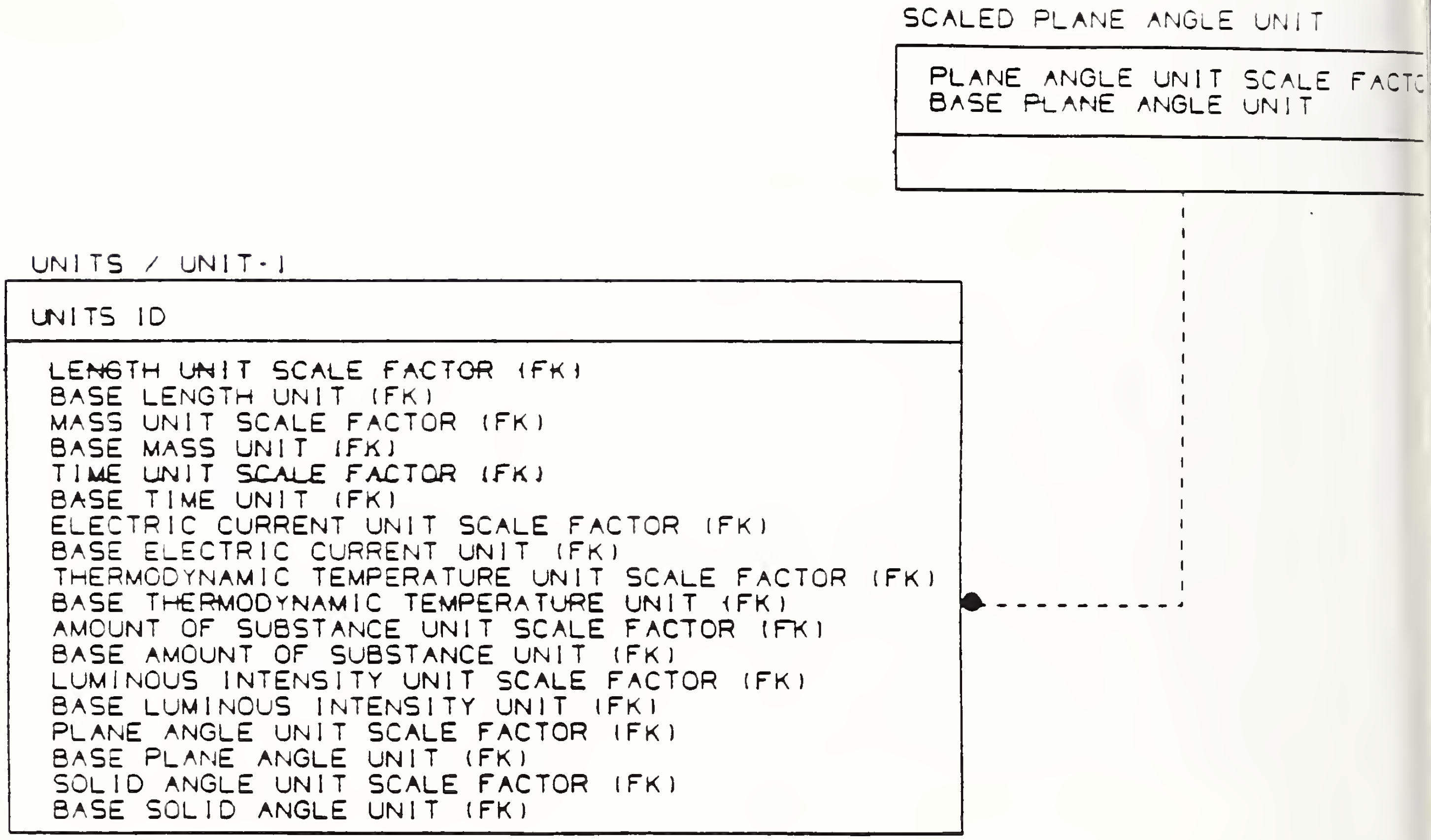

SCALED PLANE ANGLE UNIT
PLANE ANGLE UNIT SCALE FACTO
BASE PLANE ANGLE UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-158: Unit of Plane Angle

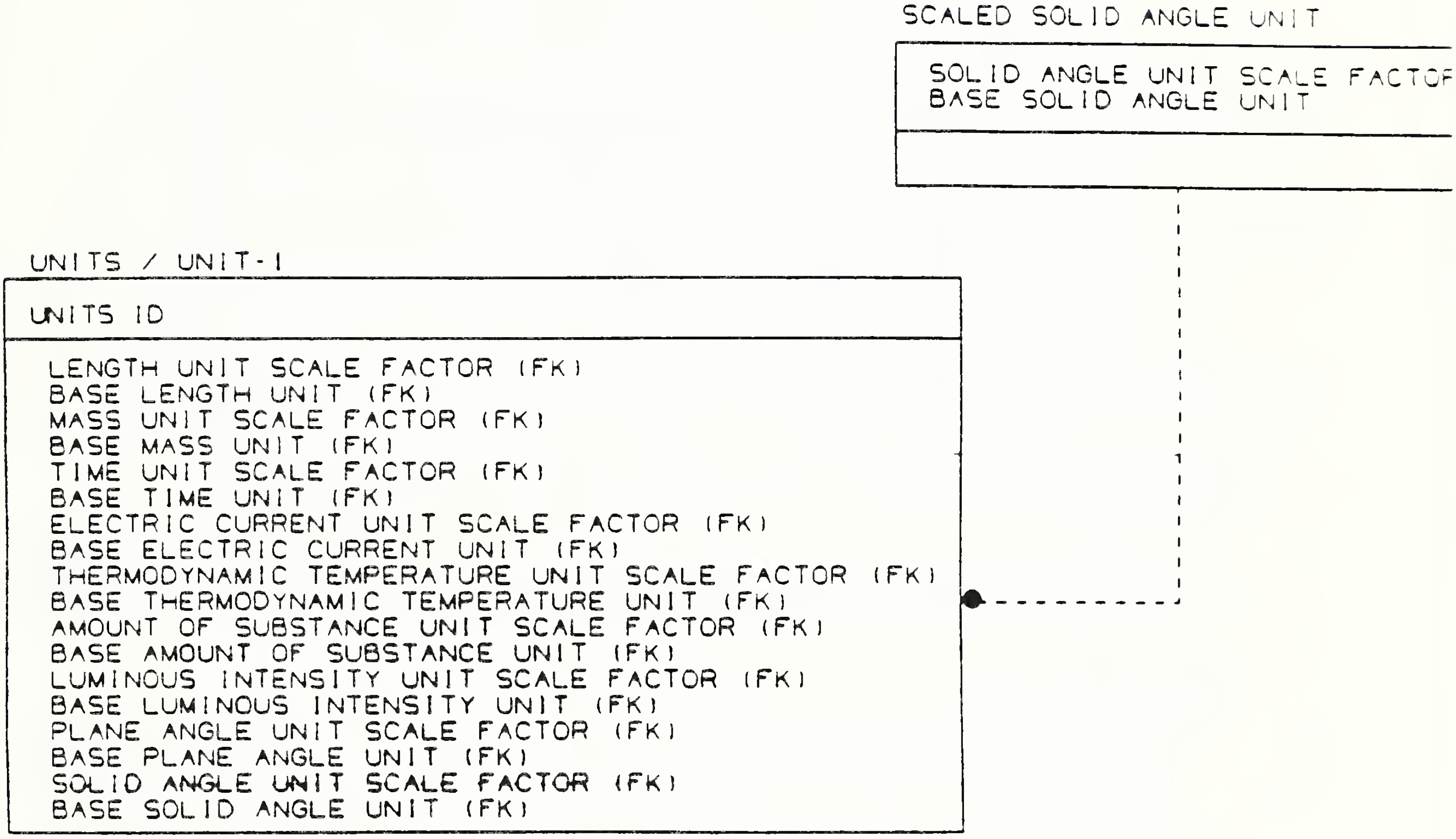

SCALED SOLID ANGLE UNIT
SOLID ANGLE UNIT SCALE FACTOR
BASE SOLID ANGLE UNIT
UNITS / UNIT-1
UNITS ID
LENGTH UNIT SCALE FACTOR (FK)
BASE LENGTH UNIT (FK)
MASS UNIT SCALE FACTOR (FK)
BASE MASS UNIT (FK)
TIME UNIT SCALE FACTOR (FK)
BASE TIME UNIT (FK)
ELECTRIC CURRENT UNIT SCALE FACTOR (FK)
BASE ELECTRIC CURRENT UNIT (FK)
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR (FK)
BASE THERMODYNAMIC TEMPERATURE UNIT (FK)
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR (FK)
BASE AMOUNT OF SUBSTANCE UNIT (FK)
LUMINOUS INTENSITY UNIT SCALE FACTOR (FK)
BASE LUMINOUS INTENSITY UNIT (FK)
PLANE ANGLE UNIT SCALE FACTOR (FK)
BASE PLANE ANGLE UNIT (FK)
SOLID ANGLE UNIT SCALE FACTOR (FK)
BASE SOLID ANGLE UNIT (FK)

Figure D-159: Unit of Solid Angle

Standard set of fundamental units of measure - British

Attributes	Values
LENGTH UNIT SCALE FACTOR	1.0
BASE LENGTH UNIT	Foot
MASS UNIT SCALE FACTOR	1.0
BASE MASS UNIT	Slug
TIME UNIT SCALE FACTOR	1.0
BASE TIME UNIT	Second
ELECTRIC CURRENT UNIT SCALE FACTOR	1.0
BASE ELECTRIC CURRENT UNIT	Ampere
THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR	1.0
BASE THERMODYNAMIC TEMPERATURE UNIT	Rakine
AMOUNT OF SUBSTANCE UNIT SCALE FACTOR	1.0
BASE AMOUNT OF SUBSTANCE UNIT	Mole
LUMINOUS INTENSITY UNIT SCALE FACTOR	1.0
BASE LUMINOUS INTENSITY UNIT	Candela
PLANE ANGLE UNIT SCALE FACTOR	1.0
BASE PLANE ANGLE UNIT	Degree
SOLID ANGLE UNIT SCALE FACTOR	1.0
BASE SOLID ANGLE UNIT	Steradian

Primary Key Attributes

UNITS ID Data Type: String

This attribute contains the unique identifier of the fundamental UNITS. There are two strings that are predefined and they are SI and British.

Other Attributes

LENGTH UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED LENGTH UNIT [Entity UNIT-2] and here is its definition:
This is a multiplicative scale factor for the BASE LENGTH UNIT attribute to define an odd or undefined length unit. For example, if a user wants a length unit of half a foot, then the BASE LENGTH UNIT would be foot and the LENGTH UNIT SCALE FACTOR would be 0,5.

BASE LENGTH UNIT Data Type: Enumeration This attribute is defined in entity SCALED LENGTH UNIT [Entity UNIT-2] and here is its definition: This attribute represents the unit of length. For example, the value of this attribute might be 'meter'. The possible values are:

- angstrom
- micron
- meter
- mil
- inch
- foot
- yard
- mile
- nautical mile
- Astronomical Unit
- Light year

MASS UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED MASS UNIT [Entity UNIT-3] and here is its definition:
This is a multiplicative scale factor for the BASE MASS UNIT attribute to define undefined mass unit. For example, if a user wants a mass unit of dekagrams, then the BASE MASS UNIT would be gram and the MASS UNIT SCALE FACTOR would be 10,0.

BASE MASS UNIT Data Type: Enumeration

See attribute definition in entity SCALED MASS UNIT [Entity UNIT-3] and here is its definition:
This attribute represents the unit of mass. For example, the value of this attribute might be 'gram'. The possible values are :

- kilogram
- tonne
- ounce
- pound mass
- slug
- drams
- carats
- grains

TIME UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED TIME UNIT [Entity UNIT-4] and here is its definition:
This is a multiplicative scale factor for the BASE TIME UNIT attribute to define an odd or undefined time unit. For example, if a user wants a time unit of leap years, then the BASE TIME UNIT would be year and the TIME UNIT SCALE FACTOR would be 1,002 74.

BASE TIME UNIT Data Type: Enumeration

See attribute definition in entity SCALED TIME UNIT [Entity UNIT-4] and here is its definition:
This attribute represents the unit of time. For example, the value of this attribute might be 'second'. The possible values are:

- second
- minute
- hour
- day
- week
- year (calendar)

ELECTRIC CURRENT UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED ELECTRIC CURRENT UNIT [Entity UNIT-5] and here is its definition: This is a multiplicative scale factor for the BASE ELECTRIC CURRENT UNIT attribute to define an odd or undefined electric current unit. For example, if a user wants an electric current unit of microampere, then the BASE ELECTRIC CURRENT UNIT would be ampere and the ELECTRIC CURRENT UNIT SCALE FACTOR would be 0,000 001.

BASE ELECTRIC CURRENT UNIT Data Type: Enumeration

See attribute definition in entity SCALED ELECTRIC CURRENT UNIT [Entity UNIT-5]and here is its definition: This attribute represents the unit of length. For example, the value of this attribute might be 'ampere'. The possible values are:

- ampere

THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED THERMODYNAMIC TEMPERATURE UNIT [Entity UNIT-6] and here is its definition:
This is a multiplicative scale factor for the BASE THERMODYNAMIC TEMPERATURE UNIT attribute to define an odd or undefined thermodynamic temperature unit. For example, if a user wants a thermodynamic temperature unit of microdegree Kelvin, then the BASE THERMODYNAMIC TEMPERATURE UNIT would be Kelvin and the THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR would be 0,000 001.

BASE THERMODYNAMIC TEMPERATURE UNIT Data Type: Enumeration

See attribute definition in entity SCALED THERMODYNAMIC TEMPERATURE UNIT [Entity UNIT-6] and here is its definition:
This attribute represents the unit of thermodynamic temperature. For example, the value of this attribute might be 'Celsius'. The possible values are :

- Kelvin
- Celsius

- Fahrenheit
- Rakine

AMOUNT OF SUBSTANCE UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity ?ALED AMOUNT OF SUBSTANCE UNIT [Entity UNIT-7] and here is its definition:

This is a multiplicative scale factor for the BASE AMOUNT OF SUBSTANCE UNIT attribute to define an odd or undefined amount of substance unit. For example, if a user wants a amount of substance unit of half a mole, then the BASE AMOUNT OF SUBSTANCE UNIT would be mole and the AMOUNT OF SUBSTANCE UNIT SCALE FACTOR would be 0,5.

BASE AMOUNT OF SUBSTANCE UNIT Data Type: Enumeration

This attribute is defined in the entity SCALED AMOUNT OF SUBSTANCE UNIT [Entity UNIT-7] and here is its definition:

This attribute represents the unit of an amount of substance. For example, the value of this attribute might be 'mole'. The possible values are:

- mole

LUMINOUS INTENSITY UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED LUMINOUS INTENSITY UNIT [Entity UNIT-8] and here is its definition:

This is a multiplicative scale factor for the BASE LUMINOUS INTENSITY UNIT attribute to define an odd or undefined luminous intensity unit. For example, if a user wants a luminous intensity unit of half a candela, then the BASE LUMINOUS INTENSITY UNIT would be candela and the LUMINOUS INTENSITY UNIT SCALE FACTOR would be 0,5.

BASE LUMINOUS INTENSITY UNIT Data Type: Enumeration

See attribute definition in entity SCALED LUMINOUS INTENSITY UNIT [Entity UNIT-8] and here is its definition:

This attribute represents the unit of luminous intensity. For example, the value of this attribute might be 'candela'. The possible values are:

- candela

PLANE ANGLE UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED PLANE ANGLE UNIT [Entity UNIT-9] and here is its definition:

This is a multiplicative scale factor for the BASE PLANE ANGLE UNIT attribute to define an odd or undefined plane angle unit. For example, if a user wants a plane angle unit of signs, then the BASE PLANE ANGLE UNIT would be degree and the PLANE ANGLE UNIT SCALE FACTOR would be 30,0.

BASE PLANE ANGLE UNIT Data Type: Enumeration

See attribute definition in entity SCALED PLANE ANGLE UNIT [Entity UNIT-9] and here

is its definition:
This attribute represents the unit of a plane angle. For example, the value of this attribute might be 'grade'. The possible values are:

- degree
- second
- minute
- radian
- revolution
- grad

SOLID ANGLE UNIT SCALE FACTOR Data Type: Real

See attribute definition in entity SCALED SOLID ANGLE UNIT [Entity UNIT-10] and here is its definition:
This is a multiplicative scale factor for the BASE SOLID ANGLE UNIT attribute to define an odd or undefined solid angle unit. For example, if a user wants a solid angle unit of solid angle, then the BASE SOLID ANGLE UNIT would be steradian and the SOLID ANGLE UNIT SCALE FACTOR would be 0,079 577 472.

BASE SOLID ANGLE UNIT Data Type: Enumeration

See attribute definition in entity SCALED SOLID ANGLE UNIT [Entity UNIT-10] and here is its definition:
This attribute represents the unit of solid angle. For example, the value of this attribute might be 'steradian'. The possible values are:

- steradian

Business rules

EXPRESS Specification

```
ENTITY units;
  scaled_length_unit : INTERNAL scaled_length_unit;
  scaled_mass_unit : INTERNAL scaled_mass_unit;
  scaled_time_unit : INTERNAL scaled_time_unit;
  scaled_electric_current_unit : INTERNAL
   scaled_electric_current_unit;
  scaled_thermodynamic_temperature_unit : INTERNAL
   scaled_thermodynamic_temperature_unit;
  scaled_amount_of_substance_unit : INTERNAL
   scaled_amount_of_substance_unit;
  scaled_luminous_intensity_unit : INTERNAL
   scaled_luminous_intensity_unit;
  scaled_plane_angle_unit : INTERNAL scaled_plane_angle_unit;
  scaled_solid_angle_unit : INTERNAL scaled_solid_angle_unit;
END_ENTITY;
```

Entity Name: **SCALED LENGTH UNIT**

Entity Number: **UNIT-2**

The SCALED LENGTH UNIT data entity defines the length unit of measure to be used.

Primary Key Attributes

LENGTH UNIT SCALE FACTOR — Data Type: Real

This is a multiplicative scale factor for the BASE LENGTH UNIT attribute to define an odd or undefined length unit. For example, if a user wants a length unit of half a foot, then the BASE LENGTH UNIT would be foot and the LENGTH UNIT SCALE FACTOR would be 0,5.

BASE LENGTH UNIT — Data Type: Enumeration

This attribute represents the unit of length. For example, the value of this attribute might be 'meter'. The possible values are:

- angstrom
- micron
- meter
- kilometer
- mil
- inch
- foot
- yard
- mile
- nautical mile
- Astronimal Unit
- Light year

Other Attributes

Business rules

EXPRESS Specification

```
TYPE length_unit = ENUMERATION OF
```

```
       (angstrom,
       micron,
       meter,
       mil,
       inch,
       foot,
       yard,
       mile'
       nautical mile,
       Astronomical Unit,
       Light year);
END_TYPE;

ENTITY scaled_length_unit;
  base_length_unit      : length_unit;
  length_unit_scale_factor      : OPTIONAL real;
END_ENTITY;
```

Entity Name: **SCALED MASS UNIT**

Entity Number: **UNIT-3**

The SCALED MASS UNIT data entity defines the mass unit of measure to be used.

Primary Key Attributes

MASS UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE MASS UNIT attribute to define undefined mass unit. For example, if a user wants a mass unit of dekagrams, then the BASE MASS UNIT would be gram and the MASS UNIT SCALE FACTOR would be 10,0.

BASE MASS UNIT Data Type: Enumeration

This attribute represents the unit of mass. For example, the value of this attribute might be 'gram'. The possible values are:

- kilogram
- tonne
- ounce
- pound
- slug
- drams
- carats
- grains

Other Attributes

Business rules

EXPRESS Specification

```
TYPE mass_unit = ENUMERATION OF
      (kilogram,
       tonne,
       ounce,
       pound,
       slug,
       drams,
       carats,
       grains);
END_TYPE;

ENTITY scaled_mass_unit;
  base_mass_unit          : mass_unit;
  mass_unit_scale_factor  : OPTIONAL real;
END_ENTITY;
```

Entity Name: **SCALED TIME UNIT**

Entity Number: **UNIT-4**

The SCALED TIME UNIT data entity defines the mass unit of measure to be used.

Primary Key Attributes

TIME UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE TIME UNIT attribute to define an odd or undefined time unit. For example, if a user wants a time unit of leap years, then the BASE TIME UNIT would be year and the TIME UNIT SCALE FACTOR would be 1,002 74.

BASE TIME UNIT Data Type: Enumeration

This attribute represents the unit of time. For example, the value of this attribute might be 'second'. The possible values are:

- second
- minute
- hour
- day

- week
- year (calendar)

Other Attributes

Business rules

EXPRESS Specification

```
TYPE time_unit = ENUMERATION OF
        (second,
        minute,
        hour,
        day,
        week,
        year);
END_TYPE;

ENTITY scaled_time_unit;
  base_time_unit         : time_unit;
  time_unit_scale_factor : OPTIONAL real;
END_ENTITY;
```

Entity Name: **SCALED ELECTRICAL CURRENT UNIT**

Entity Number: **UNIT-5**

The SCALED ELECTRIC CURRENT UNIT data entity defines the electric current unit of measure to be used.

Primary Key Attributes

ELECTRIC CURRENT UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE ELECTRIC CURRENT UNIT attribu an odd or undefined electric current unit. For example, if a user wants an electric current unit of microampere, then the BASE ELECTRIC CURRENT UNIT would be ampere and the ELECTRIC CURRENT UNIT SCALE FACTOR would be 0,000 001.

BASE ELECTRIC CURRENT UNIT Data Type: Enumeration

This attribute represents the unit of length. For example, the value of this attribute might be 'ampere'. The possible values are:

- ampere

Other Attributes

Business rules

EXPRESS Specification

```
TYPE electric_current_unit = ENUMERATION OF
        (ampere);
END_TYPE;

ENTITY scaled_electric_current_unit;
  base_electric_current_unit         : current_unit;
  electric_current_unit_scale_factor : OPTIONAL real;
END_ENTITY;
```

Entity Name: SCALED THERMODYNAMIC TEMPERATURE UNIT

Entity Number: UNIT-6

The SCALED THERMODYNAMIC TEMPERATURE UNIT data entity defines the thermodynamic temperature unit of measure to be used.

Primary Key Attributes

THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE THERMODYNAMIC TEMPERATURE UNI to define an odd or undefined thermodynamic temperature unit. For example, if a user wants a thermodynamic temperature unit of microdegree Kelvin, then the BASE THERMODYNAMIC TEMPERATURE UNIT would be Kelvin and the THERMODYNAMIC TEMPERATURE UNIT SCALE FACTOR would be 0,000 001.

BASE THERMODYNAMIC TEMPERATURE UNIT Data Type: Enumeration

This attribute represents the unit of thermodynamic temperature. For example, the value of this attribute might be 'Celsius'. The possible values are:

- Kelvin
- Celsius
- Fahrenheit
- Rakine

Other Attributes

Business_rules

EXPRESS Specification

```
TYPE thermodynamic_temperature_unit = ENUMERATION OF
        (Kelvin,
        Celsius,
        Fahrenheit,
        Rakine);
END_TYPE;

ENTITY scaled_thermodynamic_temperature_unit;
  base_thermodynamic_temperature_unit    : temperature_unit;
  thermodynamic_temperature_unit_scale_factor    : OPTIONAL real;
END_ENTITY;
```

Entity Name: SCALED AMOUNT OF SUBSTANCE UNIT

Entity Number: UNIT-7

The SCALED AMOUNT OF SUBSTANCE UNIT data entity defines the amount of substance unit of measure to be used.

Primary Key Attributes

AMOUNT OF SUBSTANCE UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE AMOUNT OF SUBSTANCE UNIT attr define an odd or undefined amount of substance unit. For example, if a user wants a amount of substance unit of half a mole, then the BASE AMOUNT OF SUBSTANCE UNIT would be mole and the AMOUNT OF SUBSTANCE UNIT SCALE FACTOR would be 0,5.

BASE AMOUNT OF SUBSTANCE UNIT Data Type: Enumeration

This attribute represents the unit of amount of substance. For example, the value of this attribute might be 'mole'. The possible values are:

- mole

Other Attributes

Business rules

EXPRESS Specification

```
TYPE amount_of_substance_unit = ENUMERATION OF
        (mole);
END_TYPE;

ENTITY scaled_amount_of_substance_unit;
  base_amount_of_substance_unit : amount_of_substance_unit;
  amount_unit_scale_factor      : OPTIONAL real;
END_ENTITY;
```

Entity Name: **SCALED LUNINOUS INTENSITY UNIT**

Entity Number: **UNIT-8**

The SCALED LUMINOUS INTENSITY UNIT data entity defines the luminous intensity unit of measure to be used.

Primary Key Attributes

LUMINOUS INTENSITY UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE LUMINOUS INTENSITY UNIT attri define an odd or undefined luminous intensity unit. For example, if a user wants a luminous intensity unit of half a candela, then the BASE LUMINOUS INTENSITY UNIT would be candela and the LUMINOUS INTENSITY UNIT SCALE FACTOR would be 0,5.

BASE LUMINOUS INTENSITY UNIT Data Type: Enumeration

This attribute represents the unit of luminous intensity. For example, the value of this attribute might be 'candela'. The possible values are:

- candela

Other Attributes

Business rules

EXPRESS Specification

```
TYPE luminous_intensity_unit = ENUMERATION OF
```

```
        (candela);
 END_TYPE;

 ENTITY scaled_luminous_intensity_unit;
   base_luminous_intensity_unit  : luminous_intensity_unit;
   luminous_intensity_unit_scale_factor  : OPTIONAL real;
 END_ENTITY;
```

Entity Name: **SCALED PLANE ANGLE UNIT**

Entity Number: **UNIT-9**

The SCALED PLANE ANGLE UNIT data entity defines the plane angle unit of measure to be used.

Primary Key Attributes

PLANE ANGLE UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE PLANE ANGLE UNIT attribute to define an odd or undefined plane angle unit. For example, if a user wants a plane angle unit of signs, then the BASE PLANE ANGLE UNIT would be degree and the PLANE ANGLE UNIT SCALE FACTOR would be 30,0.

BASE PLANE ANGLE UNIT Data Type: Enumeration

This attribute represents the unit of plane angle. For example, the value of this attribute might be 'grade'. The possible values are :

- degree
- second
- minute
- radian
- revolution

Other Attributes

vspace3ex

Business rules

vspace3ex

EXPRESS Specification

```
TYPE plane_angle_unit = ENUMERATION OF
        (second,
        minute,
        degree,
        radian,
        revolution);
END_TYPE;

ENTITY scaled_plane_angle_unit;
  base_plane_angle_unit : plane_angle_unit;
  plane_angle_unit_scale_factor : OPTIONAL real;
END_ENTITY;
```

Entity Name: **SCALED SOLID ANGLE UNIT**

Entity Number: **UNIT-10**

The SCALED SOLID ANGLE UNIT data entity defines the solid angle unit of measure to be used.

Primary Key Attributes

SOLID ANGLE UNIT SCALE FACTOR Data Type: Real

This is a multiplicative scale factor for the BASE SOLID ANGLE UNIT attribute to odd or undefined solid angle unit. For example, if a user wants a solid angle unit of solid angle, then the BASE SOLID ANGLE UNIT would be steradian and the SOLID ANGLE UNIT SCALE FACTOR would be 0,079 577 472.

BASE SOLID ANGLE UNIT Data Type: Enumeration

This attribute represents the unit of solid angle. For example, the value of this attribute might be 'sphere'. The possible values are:

- steradian

Other Attributes

Business rules

EXPRESS Specification

```
TYPE solid_angle_unit = ENUMERATION OF
        (steradian);
```

```
END_TYPE;

ENTITY scaled_solid_angle_unit;
  base_solid_angle_unit : solid_angle_unit;
  solid_angle_unit_scale_factor : OPTIONAL real;
END_ENTITY;
```

SECTION 10: UNITS INFORMATION MODEL

N288

10.4 IDEF0 Activity Model

This section is currently not in existence.

10.5 UNITS Relationship to the Planning Model

The Planning model should be shown in Figure 160. The shaded-in entity is the entity that the UNITS data model partially expands, but there is currently no Planning model in existence.

10.6 Issue Log

The following is a list for the UNITS data model. Contained in the list are:

1. An issues identifier;
2. A single character representing the current status of the issue. An R indicates that the issue is resolved. A U indicates that the issue remains unresolved. An X indicates that the issue is obsolete. An O indicates that the issue remains open and has a temporary solution defined for it, but the work remains or consensus has not been reached yet.
3. A short description

Issue	Status	Description
UNITS/Issue-1	R	Fundamental Units of Measure
UNITS/Issue-2	R	Separate Fundamental Units of Measure Entities
UNITS/Issue-3	R	Scale Factor
UNITS/Issue-4	O	Units of measure abbreviation and acronyms

Figure D-160: UNITS Planning Model

ISSUE: UNITS-1 Fundamental Units of Measure.

INITATION DATE: 16th July 1986

INITIATOR: FEM Committee

STATUS: Resolved

DESCRIPTION: There were a group of issues raised about units of measure and they are:

How does the concept of units of measurement get introduced into the data model and consequently how does it interface with the FEM data model?

Can a UNITS data entity cover all fundamental units of measure? (Is electrical current a fundamental unit of measure?)

Since units of angular measurement are not fundamental units of measure, do they need to be included within the UNITS data entity? Standard practice permits the specification of angular data in either degrees or radians. Angular position data is required for the specification of coordinate positions in either cylindrical or spherical coordinate systems. Therefore, the type of angular data must be identified. The logical place for it to be located is in a UNITS entity.

RESOLUTION: The solution chosen was to create a UNITS data entity. There are seven base units of measurement. These are length, mass, time, electric current, thermodynamic temperature, amount of substance, and luminous intensity. Also, there are two supplemental units. These are plane angles and solid angles. The UNITS data entity has one attribute for each base and supplemental unit of measurement. All floating-point values that have units must be consistently defined within the scope of this data entity.

DECISION: by ISO Subgroup 2

DECISION DATE: 1st April 1987

ISSUE: UNITS-2 Seperate Fundamental Units of Measure Entities.
INITATION DATE: 16th July 1988
INITIATOR: Various PDES committees
STATUS: Resolved
DESCRIPTION: Varies application committees have asked for different fundamental units of measure to be separated into separate entities which they may use in their data models. They do not wish to be constrained to the idea that all units in their data model are consistent and can be defined by the nine fundamental units of measure.

RESOLUTION: The solution chosen was as follows:
Create nine new entities for each of the fundamental units of measure that were defined in the UNITS entity. Those new entities are:
SCALED LENGTH UNIT
SCALED MASS UNIT
SCALED TIME UNIT
SCALED ELECTRIC CURRENT UNIT
SCALED THERMODYNAMIC TEMPERATURE UNIT
SCALED AMOUNT OF SUBSTANCE UNIT
SCALED LUMINOUS INTENSITY UNIT
SCALED PLANE ANGLE UNIT
SCALED SOLID ANGLE UNIT
Create a nonidentifing relationship for each of the new entities to the UNITS entity in order to show how the UNITS entity is made.
DECISION: by FEM Committee
DECISION DATE: 16th July 1988

ISSUE:	UNITS-3 Scale Factor.
INITATION DATE:	16th July 1988
INITIATOR:	FEM committee
STATUS:	Resolved
DESCRIPTION:	An enumeration of an unit of measure was deemed to be unable to meet the needs of the Finite Element Model (FEM) analyst for defining the units of measure he was using in a FEM analysis. The enumeration would always be incomplete set of all of the possible choices that an analyst could choose. For example, an analyst may want his unit of length to be half a foot.
RESOLUTION:	The solution chosen was to : List the common choices in the enumeration for each fundamental unit of measure. Add a scale factor for each fundamental unit of measure so that combination of the enumeration and scale factor would allow any unit of measure chosen to be described.
DECISION:	by FEM Committee
DECISION DATE:	16th July 1988

ISSUE: UNITS-4 Units of measure abbreviation and acronyms.
INITATION DATE: 24th August 1988
INITIATOR: FEM Committee
STATUS: Open
DESCRIPTION: All enumeration of an units of measure is currently is the form of complete word(s) of that unit of measure. For example, meter is meter in the length unit of measure enumeration and not M.
Should alternate spelling for units of measure, abbreviation and acronyms be allowed in the enumeration?

RESOLUTION: The current solution chosen was to not allow abbreviation and acronyms.
DECISION:
DECISION DATE:

10.7 Problem Areas and Unimplemented Ideas

This section is currently not in existence.

10.8 Related documents

This section is currently not in existence.

10.9 Complete Units SML Model

```
 VIEW UNIT-model
 GENERAL INFO
 SHORT NAME unit-model
 AUTHOR Richard Brooks
 DESCRIPTION this is the UNITS data model for version 1.0
 CREATION DATE 08/24/88
*
*       This is the Units data model.
*       (SML) input for IDEF1X modeling software
*       August 88
*
 STATUS development
 LEVEL kb
 STRUCTURE

ENTITY  UNITS
 KEY
        UNITS-ID
 ENDKEY
        SCALED-LENGTH-UNIT "has"
        SCALED-MASS-UNIT "has"
        SCALED-TIME-UNIT "has"
        SCALED-ELECTRIC-CURRENT-UNIT "has"
        SCALED-THERMODYNAMIC-TEMPERATURE-UNIT "has"
        SCALED-AMOUNT-OF-SUBSTANCE-UNIT "has"
        SCALED-LUMINOUS-INTENSITY-UNIT "has"
        SCALED-PLANE-ANGLE-UNIT "has"
        SCALED-SOLID-ANGLE-UNIT "has"
ENDENTITY

ENTITY  SCALED-LENGTH-UNIT
KEY
        BASE-LENGTH-UNIT
        LENGTH-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENTITY  SCALED-MASS-UNIT
KEY
        BASE-MASS-UNIT
        MASS-UNIT-SCALE-FACTOR
ENDKEY
```

```
ENDENTITY

ENTITY  SCALED-TIME-UNIT
KEY
        BASE-TIME-UNIT
        TIME-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENTITY  SCALED-ELECTRIC-CURRENT-UNIT
KEY
        BASE-ELECTRIC-CURRENT-UNIT
        ELECTRIC-CURRENT-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENTITY  SCALED-THERMODYNAMIC-TEMPERATURE-UNIT
KEY
        BASE-THERMODYNAMIC-TEMPERATURE-UNIT
        THERMODYNAMIC-TEMPERATURE-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENTITY  SCALED-AMOUNT-OF-SUBSTANCE-UNIT
KEY
        BASE-AMOUNT-OF-SUBSTANCE-UNIT
        AMOUNT-OF-SUBSTANCE-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENTITY  SCALED-LUMINOUS-INTENSITY-UNIT
KEY
        BASE-LUMINOUS-INTENSITY-UNIT
        LUMINOUS-INTENSITY-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY


ENTITY  SCALED-PLANE-ANGLE-UNIT
KEY
        BASE-PLANE-ANGLE-UNIT
        PLANE-ANGLE-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY
```

```
ENTITY  SCALED-SOLID-ANGLE-UNIT
KEY
        BASE-SOLID-ANGLE-UNIT
        SOLID-ANGLE-UNIT-SCALE-FACTOR
ENDKEY
ENDENTITY

ENDS
ENDVIEW
```

10.10 Complete Units EXPRESS Model

The FEM committee of ISO TC184/SC4/WG1 have been developing their reference model using the graphical data modelling language IDEF1X. This section contains a translation of that reference model from IDEF1X to EXPRESS. It is used a basis for the integration of UNIT into the Integrated Product Information Model.

There is no algorithm available for direct translation from IDEF1X to EXPRESS, and some of the IDEF1X entities had several options for modeling in EXPRESS. In the cases where it was not clear which solution was the best, notes are given which explain the difficulty. These can be found in note section at the end of the EXPRESS listing.

The following page contains a list of general rules which were followed wherever possible during the translation. They are rules which seemed to give a sensible EXPRESS model for many IDEF1X entities (but it was not possible to apply them universally).

10.10.1 General Rules

1. Every IDEF1X entity has a corresponding EXPRESS entity unless the IDEF1X entity exists only to resolve a many-to-many relationship between two other entities. In this case the IDEF1X entity will usually appear only as an attribute of one of the two other entities. An IDEF1X entity may also not have a corresponding EXPRESS entity under rule 6 below.

2. Where an IDEF1X entity has a foreign key attribute above the line and the relationship between this entity and its parent is 1 or many, then this key will not appear as an attribute of the entity in EXPRESS but the relationship is satisfied by the entity being referenced by a attribute of type set in the parent entity.

3. Where an IDEF1X entity has a foreign key attribute above the line and the relationship between this entity and its parent is 0, 1 or many, then this may appear either as an attribute in the child entity or as a set of entity references in the parent entity.

4. Where an IDEF1X entity has a foreign key attribute below the line this appears as an entity reference attribute in the corresponding EXPRESS entity.

5. Each IDEF1X entity, except those mentioned in 6 below, has one key attribute (above the line) which is not a foreign key. This will generally not appear in EXPRESS unless it is considered an important piece of information which it is desired to access externally.

6. Where there is a 0 or 1 relationship between entities then the child can normally be considered as an optional attribute of the parent and will appear as such in the EXPRESS model.

10.10.2 UNITS Reference Model

```
SCHEMA Units_measure;

TYPE length_unit = ENUMERATION OF
        (angstrom,
        micron,
        millimetre,
        metre,
        kilometre,
        mil,
        inch,
        foot,
        yard,
        mile);
END_TYPE;

TYPE mass_unit = ENUMERATION OF
        (milligram,
        gram,
        kilogram,
        tonne,
        ounce,
        pound,
        ton);
END_TYPE;

TYPE time_unit = ENUMERATION OF
        (second,
        minute,
        hour,
        day,
        week,
        year);
END_TYPE;

TYPE current_unit = ENUMERATION OF
        (ampere);
END_TYPE;

TYPE temperature_unit = ENUMERATION OF
        (kelvin,
        celsius,
        fahrenheit);
END_TYPE;

TYPE amount_of_substance_unit = ENUMERATION OF
        (mole);
END_TYPE;

TYPE luminous_intensity_unit = ENUMERATION OF
        (candela);
END_TYPE;

TYPE plane_angle_unit = ENUMERATION OF
        (second,
        minute,
        degree,
        radian);
END_TYPE;

TYPE solid_angle_unit = ENUMERATION OF
        (steradian);
END_TYPE;

ENTITY units;
  scaled_length_unit    : INTERNAL scaled_length_unit;
  scaled_mass_unit      : INTERNAL scaled_mass_unit;
  scaled_time_unit      : INTERNAL scaled_time_unit;
  scaled_electric_current_unit  : INTERNAL scaled_current_unit;
  scaled_thermodynamic_temperature_unit : INTERNAL scaled_temperature_unit;
  scaled_amount_of_substance_unit       : INTERNAL scaled_amount_of_substance
  scaled_luminous_intensity_unit        : INTERNAL scaled_luminous_intensity_
  scaled_plane_angle_unit       : INTERNAL scaled_plane_angle_unit;
  scaled_solid_angle_unit       : INTERNAL scaled_solid_angle_unit;
END_ENTITY;

ENTITY scaled_length_unit;
  base_length_unit      : length_unit;
  length_unit_scale_factor      : OPTIONAL real;
END_ENTITY;

ENTITY scaled_mass_unit;
  base_mass_unit        : mass_unit;
  mass_unit_scale_factor        : OPTIONAL real;
END_ENTITY;
```

```
ENTITY scaled_time_unit;
  base_time_unit          : time_unit;
  time_unit_scale_factor          : OPTIONAL real;
END_ENTITY;

ENTITY scaled_current_unit;
  base_electric_current_unit      : current_unit;
  current_unit_scale_factor       : OPTIONAL real;
END_ENTITY;

ENTITY scaled_temperature_unit;
  base_thermodynamic_temperature_unit   : temperature_unit;
  temperature_unit_scale_factor : OPTIONAL real;
END_ENTITY;

ENTITY scaled_amount_of_substance_unit;
  base_amount_of_substance_unit : amount_of_substance_unit;
  amount_unit_scale_factor        : OPTIONAL real;
END_ENTITY;

ENTITY scaled_luminous_intensity_unit;
  base_luminous_intensity_unit  : luminous_intensity_unit;
  luminous_intensity_unit_scale_factor  : OPTIONAL real;
END_ENTITY;

ENTITY scaled_plane_angle_unit;
  supplementary_plane_angle_unit        : plane_angle_unit;
  plane_angle_unit_scale_factor : OPTIONAL real;
END_ENTITY;

ENTITY scaled_solid_angle_unit;
  supplementary_solid_angle_unit        : solid_angle_unit;
  solid_angle_unit_scale_factor : OPTIONAL real;
END_ENTITY;

END_SCHEMA;
```

10.10.3 Notes

None.

Title: Development Support

Owner: Jeff Altemueller

Date: 10 October 1988

Corresponding ISO Document Number: N289

N289

DEVELOPMENT SUPPORT

DOCUMENT NUMBER : ANNEX D (Informative) VERSION : 1

TITLE : DEVELOPMENT SUPPORT

ABSTRACT :

The purpose of this document is to provide the STEP software developer with relevant information on STEP implementations, helpful software tools, sample test cases, etc.. This will be a "living" document that will continue to grow as STEP implementation experience grows.

KEYWORDS : Implementation guide, Software Tool, Test Case

DATE : OCTOBER 10, 1988

OWNER : Jeff Altemueller
McDonnell Douglas Corp.
Dept. W315, Bld 271, MS298
P.O. Box 516
ST. Louis, MO. USA

TELEPHONE : 314 234 5272
TELEFAX : 314 777 1704

ISO REPRESENTATIVE:
Jeff Altemueller

STATUS : Working Paper

TABLE OF CONTENTS

D.1 STEP IMPLEMENTATION GUIDE

D.2 LIBRARY OF STEP TEST PARTS

D.3 SOFTWARE TOOLS

D.4 SPECIAL TREATISES

1 STEP Implementation Guide

Table of Contents

1. Introduction..

2. STEP Levels of Implementation...................................
 2.1 Purpose...
 2.2 Categories of Criteria..
 2.3 The General Definitions.......................................

3. STEP Exchange Format File Implementations.......................

 3.1 PDDI System Translator..
  3.1.1 System Architecture..
  3.1.2 Design Trade-offs..
  3.1.3 Detail Design..
   3.1.3.1 Pre-Processor...
   3.1.3.2 Post-Processor..
   3.1.3.3 Error Recovery..
  3.1.4 PDDI Data Dictionary.......................................
 3.2 Error Recovery..
  3.2.1 Categories of Errors.......................................
  3.2.2 General Recovery Strategies................................
 3.3 Data Dictionaries...
  3.3.1 Structure..
  3.3.2 Access...
  3.3.3 Capabilities...

4. STEP Working Form Implementations...............................

7.7.1 From a 3D 3-Point Circular Arc..
7.7.2 To a 3D 3-Point Circular Arc..
7.7.3 To a 5-Point Conic Arc..
7.8 Bezier_Curve
7.8.1 From a Single Segment Hermitian Parametric Cubic Curve.........
7.8.2 To a Single Segment Hermitian Parametric Cubic Curve...........
7.9 Plane
7.9.1 From a Point, Vector Plane...
7.9.2 To a Point, Vector Plane..
7.10 Spherical_Surface
7.10.1 From a Center Point, Vector Sphere...............................
7.10.2 To a Center Point, Vector Sphere..................................
7.11 Conversion Issues...

8. Compatibility with other standards..
8.1 ISO 6937 ASCII Character Encoding...................................
8.2 ISO 8859 ASCII Character Encoding...................................
8.3 ISO 8211 Data Descriptive Files..
8.4 ISO Open Systems Interconnect Model................................
8.4.1 FTAM...
8.5 Abstract Syntax Notation 1 (ASN.1).................................
8.6 MAP / TOP...
8.7 ODA / ODIF..

4.1 PDDI Working Form ....
4.1.1 PDDI Model Access Software ....

5. STEP Data Base Implementations ....
5.1 Hierarchical ....
5.2 Network ....
5.3 Relational ....
5.4 Objected-oriented ....

6. STEP Software Binding ....
6.1 Specific Entity Bindings ....
6.1.1 Parametric Evaluators ....
6.2 Generic Entity Bindings ....

7. Geometric Conversion Algorithms ....
7.1 Notational Conventions ....
7.2 Predefined Functions ....
7.3 Cartesian_Three_Coordinate
7.3.1 From a 3D Cartesian Point ....
7.3.2 To a 3D Cartesian Point ....
7.4 Line_Segment
7.4.1 From a 3D Bounded Line ....
7.4.2 To a 3D Bounded Line ....
7.5 Circle
7.5.1 From a 3D, 2-Point, Normal Vector Circle ....
7.5.2 To a 3D, 2-Point, Normal Vector Circle ....
7.6 Ellipse
7.6.1 From a 3D 3-Point Ellipse ....
7.6.2 To a 3D 3-Point Ellipse ....
7.7 Trimmed_Curve

1.0 INTRODUCTION

The purpose of this implementor's guide is to give the STEP software developer an understanding of STEP as it relates to specific types of software implementations. This guide will offer suggestions to the developer on such topics as translator architecture, parser design, error recovery, and common geometric conversion algorithms. This will be a "living" document that will branch out into other aspects of implementation as they are explored.

The information presented here is collected from a variety of sources. Since each source may deem different aspects of their implementation proprietary, the completeness of each implementation discussion may vary. Despite this inconsistency, no contributions to this document are thought to be without merit. Contributions to this document are highly encouraged from its readership.

It is sincerely hoped that STEP software developers will both benefit from, and contribute to, the ideas and experience presented in this document.

2.0 STEP IMPLEMENTATION LEVELS

2.1 PURPOSE

The purpose of establishing implementation levels is to allow system developers to improve the data exchange process between systems. The data exchange process can be optimized when system developers are able to understand or classify the architectures of the various systems that they must integrate. In order to usefully classify implementations, system developers need to develop a common terminology and method for doing so. That is the purpose of this section.

The classification of a STEP implementation into a level or a set of levels will be performed by comparing the implementation against a specific set of criteria. At this time, there are nine areas or categories of criteria

There will probably be more than nine categories of criteria, and most categories will certainly require more than one question in order to be adequately covered. The refinement of these categories of criteria will be an ongoing process, performed as our experience grows. It is intended for these categories to serve as a starting point for classifying implementations The categories are listed below.

2.2 CATEGORIES OF CRITERIA

Category #1

"Can the implementation read / write to the standard exchange format file"

__ yes
__ no

Category #2

"Can the system generate / access the standard working form ?"

__yes
__no

Category #3

"What is the granularity of the accessable data in this implementation ?

__can query on a single entity attribute
__can query on multiple entity attributes
__can query on a single entity
__can query on multiple entities
__can query on a single model
__can query on multiple models

Category #4

"What standard forms of a data query are supported for this implementation ?

__No standard query form supported
__Standard function calls only
__Standard interactive DML only
__Standard batch DML only
__All standard forms of query

Category #5

"What standard forms of a data query response are supported for this application ?"

__No standard forms of data query response are supported
__Standard function's output parameters only
__Standard reply to a file
__Standard reply to an output device
__All standard forms of reply

Category #6

"What physical locations for the data can the implementation support ?"

__In local computer memory
__On local computer disk
__On local and/or remote computers

Category #7

"What are the logical navigational capabilities of the implementat against the IPDM ? "

__Hierarchical queries supported
__Network queries supported
__Relational queries supported
__Other ______________

Category #8

"What are the constraint checking capabilities of the implementation ?"

IF
__No constraint checking
__Some constraint checking
__All constraint checking
WHAT
__Simple constraint checking
__Complex constraint checking
HOW
__Performed by hard-coded function calls
__Performed by executing external directives
__Performed by other means
WHEN
__Performed once when data is loaded into system

__Performed each time data is used

Category #9

"What is the automation level of the data sharing process ?"

__Not automated (i.e. manual only)
__Partially automated (e.g. automatic notification, some manual ste
__Fully automated (updates occur automatically, always current data

It is important for the reader to understand that these nine categories do not imply that there are necessarily nine implementation leve It must also be apparent to the reader that several of the categories of criteria are not mutually exclusive. In other words, it will be possible fo implementations to be classified identically under some categories and differently under others. This fact does not reduce the potential usefulnes of classifying implementations. Implementations do not need to match in eve category in order to effectively exchange data.

2.3 THE GENERAL DEFINITIONS

Previous discussions have made it clear that PDES implementations are classifiable in too many categories for there to be exactly four comple definitions. However, even if they are not fully comprehensive, it can be useful to have a handful of rough definitions. Therefore, generalized definitions of the implementation levels are listed below.

LEVEL 1 FILE EXCHANGE

Product data is translated into or out of the standard exchange forma file using non-standard software. The data in the exchange file is derived the IPDM. The granularity of the data in the exchange file will range from multi-model level to the entity level. No standard forms of query are defin for the product data when in this form. No standard navigational capabiliti are defined for the product data when in this form. No standard validity constraint checks are defined for the product data when in this form. The automation level of the data sharing process is not defined when the produc data is in this form.

LEVEL 2 WORKING FORM EXCHANGE

Product data is translated into or out of the standard working form. The product data in the working form is derived from the IPDM. The granularity of the data in the working form will range from the model level to the entity level. The working form must be accessable by standard exchange format files or standard access software function calls. The worki form must support relational, network, and hierarchical types of queries

LEVEL 3 DATABASE EXCHANGE

Product data is translated into or out of a Database Management System (DBMS). The product data in the DBMS is derived from the IPDM. The granularity of the data in the DBMS will range from the multi-model level to the entity level. The DBMS must be accessable by standard exchange format files, standard access software function calls, or standard Data Manipulation Language statements. The DBMS must support relational, network, and hierarchical types of queries against the product data. The DBMS will not enforce validity constraints. The data sharing process is fully automated when the product data is in this form.

LEVEL 4 KNOWLEDGEBASE EXCHANGE

Product data is translated into or out of a Knowledgebase Management System (KBMS). The product data in the KBMS is derived from the IPDM. The granularity of the data in the KBMS will range from the multi-model level to the entity level. The KBMS must be accessable by standard exchange format files, standard access software function calls, or standard Data Manipulation Language statements. The KBMS must support relational, network, and hierarchical types of queries against the product data. The KBMS will enforce all of the validity constraints specified in the standard. The data sharing process is fully automated when the product data is in this form.

3.0 STEP Exchange Format Implementations

This chapter will contain descriptions of a variety of STEP software implementations that involve the reading or writing of STEP exchange format files. These implementations may also involve some type of Working Form or Database implementation as well, if so, those aspects of the implementation will be covered in more detail in later chapters.

3.1 THE GMAP SYSTEM TRANSLATOR

The presentation of materials describing this translator has been delayed by the continued classification of that documentation by the United States Air Force as "For Early Domestic Dissemination Only" (FEDD). Upon the removal of the FEDD clause from those documents, a detailed description of that translator will be included here.

4.0 STEP WORKING FORM IMPLEMENTATIONS

This chapter will contain descriptions of a variety of STEP software implementations that involve a working form and at least some types of software that provide access to it. These implementations may also involve some type of Database implementation as well, if so, those aspects of the implementation will be covered in more detail in later chapters.

4.1 THE PDDI ACCESS SOFTWARE AND WORKING FORM

The following pages have been excerpted from the PDDI ACCESS SOFTWARE USER MANUAL. The complete document can be obtained by contacting ???? and asking for document ????.

3.7 Model Access Software

The PDDI Access Software is a set of Pascal procedures that maintains the physical structure of related user data in computer memory. This user data is referred to as the working form model. The package provides an interface to the working form model for application programs to create, relate, and access elements of user data.

The application programs are independent of the physical structure of the stored data elements. This independence ensures that as different structure techniques are implemented, the application programs need not change.

This package manages two types of data: entities and lists. An entity is an element of data supplied by the application to be stored in the working form. A list is a collection of entity keys. The package manages lists created by the application in the working form.

The Access Software allows the structuring of the user data. The entities can be related in user/constituent order. An entity may be related to multiple user entities, creating a network structure in the working form. An entity may also contain multiple constituent entities.

3.8 DATA ITEMS

The Access Software manages two types of data items within the working form - Entities and Lists.

ENTITY

An entity is the principle data item managed by the Access Software, and is:

- Defined by the conceptual schema in the application creating the entity.
- Accessed by a unique key return from the create entity function
- A node in the working form structure containing an Attribute Data Block(ADB), and references to other entities in Constituent Relationships and/or User Relationships

ATTRIBUTE DATA BLOCK

An Attribute Data Block(ADB) is a collection of data embedded in a single contiguous block of memory. Individual pieces of data within an ADB are call attributes. MAS manages only the first three items in the structure of an ADB. These three attributes, KIND, LENGTH, and SYSUSE, are required in every entity. A short description of each attribute follows:

KIND - Must be the first item defined in the ADB. The KIND defines the entity type code. This code cannot be changed.

LENGTH - Must be the second item defined in the ADB. The LENGTH defines the number of bytes in the ADB including KIND, LENGTH, and SYSUSE.

SYSUSE - One full word of system use data reserved for internal purposes. This data is never used by the application, and should never be inspected or modified.

NOTE: All other attribute data in the ADB is managed by the application program.

CONSTITUENT RELATIONSHIP

A constituent entity is used in the definition of the user entity. Inclusive constituents of an entity encompass all descendents, their descendents, and so forth until there are no more descendents. For example in Figure 3-4, Point 0 (P0) and Point 1 (P1) are constituents of Line 1.

```
LINE = ENTITY(5008);
  IDENT  : KEY T_IDENT;
  DISPLA : T_DISPLAY;
  P0     : POINT;
  P1     : POINT;
  END_ENTITY;
```

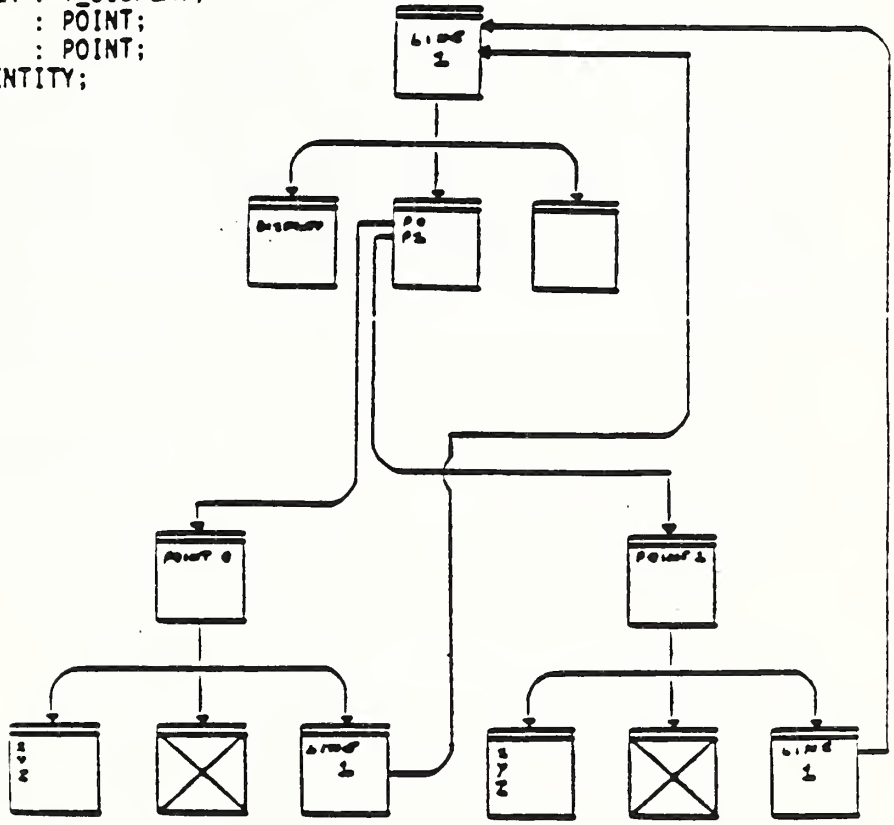


Figure 3-4 LINE: An Entity With Constituents

USER RELATIONSHIPS

A user entity uses constituent entities in its definition. Inclusive users of an entity include all ancestors, their ancestors, and so forth until there are no more ancestors. For example in Figure 1-1, Line 1 is a user of Point 0 (P0) and Point 1 (P1).

LIST

A list is a collection of entity keys which is:

- Created by the Application program.
- Accessed by a unique list key returned from the Create List Functions.
- Used by the Application to store selected entity keys for subsequent processing.

3.9 INTERFACE PARAMETERS

Each interface parameter has a name and a type. This information is shown a; follows:

DATA-NAME:DATA-TYPE.

DATA-NAME PARAMETERS

The following conventions are used to name parameters:

- Keys are named KEY1, KEY2,...KEYN.
- The ADB is named ENTDEF.
- Text parameters are named according to their purpose.
- Integer parameters are named according to their purpose.
- A return code is produced by every interface routine/operation. This parameter is a full word integer and is always named IRC. (See Appendix for a return code list.)

DATA-TYPE PARAMETERS

Data-Type parameters may be one of the following:

ANYKEY - Access key of an entity or list.

ENTBLOCK - Entity data block definition.
- In Pascal, probably declared as a record.
- In Fortran, declared as a common or dimension array.

CHARACTER - A single character as defined by the system.

INTEGER - A full word integer.

FORMAL DATA TYPES

The following is a reference list of data-types for interface calls in this MAS document.

```
         ANYKEY       = INTEGER
         LISTKEY      = ANYKEY
         ENTKEY       = ANYKEY
         ORD_KIND     = INTEGER
         EXT_RET_CODE = INTEGER
         LISTPSTN     = INTEGER
         LISTINDX     = INTEGER
         LISTSIZE     = INTEGER
         ROUTINE      = ARRAY(1...8) OF CHARACTER
         NAMTYP       = ARRAY(1...6) OF CHARACTER

(ADB)    ENTBLOCK     = RECORD OF
           KIND         = ORD_KIND
           SIZE         = INTEGER
           SYSUSE       = INTEGER
           DATA         = (USER DEFINED)
```

PASCAL APPLICATIONS

The formal declarations for the Access Software interface routines are maintained in the member APL TYP of the library "CADS.FRMI.MASyymmdd.INCLD"

Where:
y = year
mm = month
dd = day
of the latest Access Software release.

3.10 MEMORY MANAGER

A Model Access Memory Manager was developed to replace the PASCAL run-time memory manager. It reduces the number of bytes of overhead required for free-space collection, and isolates the working form model from all other PASCAL dynamic allocations.

This memory manager is currently in the MAS package and requires no user intervention for utilization.

SECTION 4

INTERFACE OPERATIONS

Page

INTRODUCTION 4-3

INITIALIZATION/DELETION OF THE MAS ENVIRONMENT 4-4
    MAINIT. 4-5
    MAKILL. 4-6

ENTITY OPERATIONS 4-7

CREATE OPERATIONS 4-8
    MAECR. 4-9
    MAEC 4-10
    MAECI. 4-11
    MAECIK 4-12
    MAEU 4-13
    MAEUI. 4-14
    MAEUIK 4-15
GET OPERATIONS. 4-16
    MAEGKN 4-17
    MAEGTK 4-18
UPDATE OPERATIONS 4-19
    MAEUD. 4-20
DELETE OPERATIONS 4-21
    MAED 4-24
    MAEDI. 4-25
    MAEDT. 4-26
    MAEDTI 4-27
    MAEDTS 4-28
ACTIVATE OPERATIONS 4-29
    MAEA 4-30
    MAEAI. 4-31
    MAEAV. 4-32
APPLICATION FLAG OPERATIONS 4-33
    MAERST 4-34
    MAQURY 4-35
    MAUPDT 4-36
    MAECQY 4-37
    MAECMP 4-38
    MAESWA 4-39
    MAESWT 4-40
    MAESVL 4-41

INTERFACE OPERATIONS (CONTINUED)

Page

LIST OPERATIONS 4-42

BOOLEAN OPERATIONS 4-43
 MALAND. 4-44
 MALNOT. 4-45
 MALOR 4-46
STRUCTURE OPERATIONS 4-47
 MALK. 4-48
 MALKL 4-49
GENERAL OPERATIONS 4-50
 MAL 4-51
 MALN. 4-52
 MALCPY. 4-53
 MALFND. 4-54
 MALNO 4-55
 MALGTK. 4-56
DELETE OPERATIONS. 4-57
 MALD. 4-58
 MALDA 4-59
 MALDI 4-60
 MALOCK. 4-61
EDIT OPERATIONS. 4-62
 MALATC. 4-63
 MALINS. 4-64
 MALRDE. 4-65
 MALREP. 4-66
 MALRMV. 4-67
 MALROR. 4-68
 MALRPL. 4-69
 MALRVS. 4-70
 MALSRT. 4-71
SEQUENTIAL READ AND EXECUTE OPERATIONS 4-72
 MALRD 4-73
 MALSTF. 4-74
 MALSTR. 4-75
 MAEXEQ. 4-77
 MAKXEQ. 4-78
 MALXEQ. 4-79
 MAECXQ. 4-81
 MAEUXQ. 4-82

4.1 INTRODUCTION

The Entity Operations and List Operations sections provide the applications programmer with the interface operations needed to access the data structures passed back to the application program. (See appendix for Pascal and FORTRAN Schema Diagrams.)

Figure 4-1 illustrates the interrelationships of the Access Software interface operations shown in these sections.

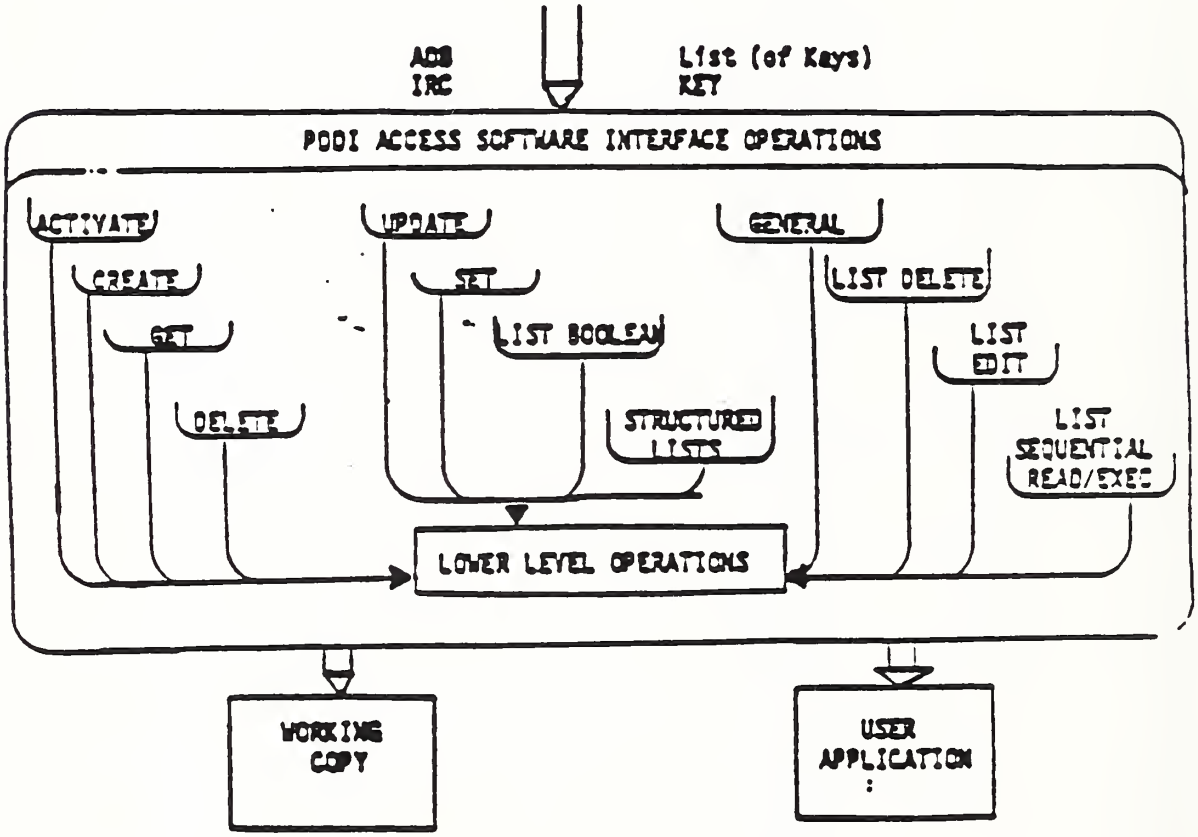


Figure 4-1 Interface Operations

4.2 INITIALIZATION/DELETION OF THE MAS ENVIRONMENT

Two routines provide the interface used to initialize the Access Software.

The basic initialization operation (MAINIT) creates a working model and enables the Access Software.

The MAKILL function is used to destroy the working model and disable the Access Software.

An application does not have to install a data dictionary. It can create and use entities on an *ad hoc* basis. If a data dictionary is not installed, the following limitations are imposed:

1. Any entity type will be permitted.
2. The interface routines will not validate any operation other than outright errors; e.g., defining an ADB with a negative length. The application is - "on its own".
3. There will be no provision for organization of entities by class.

Included with the initialization and deletion operations descriptions that follow are the error and warning messages that may be returned. Appendix A contains a complete list of these messages along with their numeric codes.

4.3 ENTITY OPERATIONS

The basic entity operations can be categorized by the following functions:

Activate
Create
Get
Delete
Update
Process Flags
Application Flags

All operations performed on entity constituent lists are done by list operations, with the exception of creating an entity with constituents.

Included with the entity operations descriptions presented on the pages that follow are the error and warning messages that may be returned. Appendix A contains a complete list of these messages along with their numeric codes.

4.3.1 CREATE OPERATIONS

These operations allow the creation of entities in the working model. The application creates the entity in its local memory space. This includes the specification of KIND, LENGTH, and any other attribute data as needed. The KIND value cannot change. The LENGTH value can be changed by the MAEUD function.

The create routines are shown in the following table.

DESCRIPTION	ROUTINE
Create an entity.	MAECR
Create an application list of constituent entity references.	MAEC
Create an application list of inclusive constituent entities.	MAECI
Create an application list of inclusive constituents by KIND.	MAECIK
Create an application list of user entity references.	MAEU
Create an application list of inclusive user entities.	MAEUI
Create an application list of inclusive users by KIND.	MAEUIK

4.3.2 GET OPERATIONS

These operations are used to get the MAS copy of a specified entity attribute block and load it into the application memory area. Get operations are also used to get a specified attribute in the entity ADB.

The get routines are shown in the table below.

DESCRIPTION	ROUTINE
Get the KIND value of a specific entity.	MAEGKN
Get the ADB of a specific entity.	MAEGTK

4.3.3 UPDATE OPERATIONS

These operations are used to update the ADB for specified entities. In general, the application should use the MAEGTK function to get the ADB before the update function is used.

The update routine is shown in the following table.

DESCRIPTION	ROUTINE
Update the attribute data block of an entity.	MAEUD

4.3.4 DELETE OPERATIONS

These operations address how you delete entities from the MAS working form model. The entities in the working model currently are grouped into the following classifications:

- Dependent
- Support
- Primary
- Secondary

Delete rules have been established for the entities in these classifications. For a new entity kind, the default classification is "Dependent" unless it is otherwise defined.

DELETE RULES

The delete rules apply to the constituent relationships with which entities are defined. They determine whether a constituent entity can be deleted by checking each of its user entities. For example, the delete rules applied to entity (A) in relation to a specific user entity (B) may be different than the delete rules for that same entity (A) in relation to another specific user entity (C).

The action taken for the IDB/MAS delete classifications are determined by the combinations of yes/no (Y/N) answers to the following conditions, posed as questions:

(1) Can this constituent entity be deleted from a specific user entity?

(2) Does the deletion of this (constituent) entity cause deletion of a specific user?

(3) Does deletion of a specific user cause deletion of this entity (constituent)?

CONDITION			
(1)	(2)	(3)	DELETE CLASSIFICATION
N	N	N	Dependent
N	N	Y	Support
N	Y	N	Primary
Y	N	N	Secondary

The delete classifications are defined as follows:

Dependent - Constituent entity cannot be deleted because the user entity is dependent on its existence. The user entity may be deleted without deleting the constituent entity.

Support - Constituent entity cannot be deleted because the user entity is dependent on its existence. The user entity may be deleted; however, the constituent entity will also be deleted unless another user entity does not permit the deletion of the constituent entity.

Primary - Constituent entity can be deleted, but only if the user entity can, and will, also be deleted. The user entity may be deleted without the constituent entity being deleted.

Secondary - If the number of constituents falls below an established minimum, the constituent entity can be deleted and, if possible, the user entity will also be deleted. If the user entity cannot be deleted, none of the minimum constituents can be deleted. If the number of constituents is greater than or equal to the minimum, the constituent entity can be deleted.

Test routines are provided to return the entities or lists that would be deleted if actual delete routines were used.

DELETE ROUTINES

The IDB/MAS delete routines are presented in the following table. The first two routines actually delete entities (MAED, MAEDI). The third and fourth routines test the delete function, allowing the programmer to see the results of a potential delete without modifying the stored data (MAEDT, MAEDTI).

When deleting a list of entities that includes users and constituents, the list should be ordered so that the users are processed before the constituents. The routines MALROR and MALRORI perform this function on an application list. (An entity constituent list should never be reordered.)

DESCRIPTION	ROUTINE
Delete an entity or list of entities.	MAED
Delete an entity or list of entities and the inclusive constituents.	MAEDI
Delete test an entity or list of entities.	MAEDT
Delete test an entity or list of entities and the inclusive constituents.	MAEDTI

4.3.5 ACTIVATE OPERATIONS

These operations are used to activate an entity. An entity is deactivated when a delete was attempted, but was not completed because of the user's dependency condition on the entity. (See Delete Operations Section)

The activate routines are shown in the table below.

DESCRIPTION	ROUTINE
Activate an entity or list of entities.	MAEA
Activate an entity or list of entities and their inclusive constituents.	MAEAI
Find the present value of the activation setting for an entity.	MAEAV

NOTE:

- Activation is not the same as rejection after a delete. If an entity was deleted, then it cannot be recovered with these functions.
- Activation functions will activate any entity regardless of when or how it was made inactive.

4.3.6 APPLICATION FLAG OPERATIONS

These operations are used to get or set any application accessible fla associated with an entity.

The Application Flag routines are shown in the following table.

DESCRIPTION	ROUTINE
Reset any application accessible flag for all entities in the model.	MAERST
Determine the value of a given application accessible flag of an entity.	MAQURY
Update the value of a given application accessible flag of an entity or list of entities.	MAUPDT
Determine whether the user compresses with its constituent.	MAECQY
Create a list of constituents with which the input entity compresses.	MAECMP
Reset Process Flag for all entities in the model.	MAESWA
Set the Process Flag in an entity or list of entities.	MAESWT
Find the Process Flag setting of an entity.	MAESVL

4.4 LIST OPERATIONS

This section explains the use of the MAS list operations. A list is a temporary internal structure that contains references to entities. Since the application can build lists that take up space in the working model, it is necessary that the applications periodically delete the lists that are no longer needed.

Many list operations will accept either a list key or an entity key as input keys. When an entity key is supplied, it is assumed that the constituent list of the entity becomes the list to be operated on.

Some operations on lists may result in the same entity being in the output list more than once. The operation (MALRDE) can be used to remove duplicate entities from the list.

All operations that create an application list automatically set the position of the list to the beginning. Once a list has been read to the end, it must be reset before the sequential read process can begin again.

When an entity is deleted, all references to it in all application lists are automatically removed and the current positions of the affected lists are adjusted to retain their original meaning.

The basic list operations can be categorized by the following functions:

Boolean
Structure
General
Delete
Edit
Sequential Read
Execute

Included with the list operations descriptions are the error and warning messages that may be returned. Appendix A contains a complete list of these messages along with their numeric codes.

4.4.1 BOOLEAN OPERATIONS

For Boolean operations, there are two input lists and one output list. The application is responsible for providing two input lists consisting with the Boolean operation to be performed. No validation checking is done. If one or both of the input lists contain duplicate entities, then the output list may also contain duplicate entities. This result may not be consistent with the Boolean theory operation being performed.

The Boolean routines are shown in the following table.

DESCRIPTION	ROUTINE
Create a list from a Boolean "AND" on two input lists.	MALAND
Create a list from a Boolean "NOT" on two input lists.	MALNOT
Create a list from a Boolean "OR" on two input lists.	MALOR

4.4.2 STRUCTURE OPERATIONS

The following table presents the structure routines:

DESCRIPTION	ROUTINE
Create a list of entities with a given KIND.	MALK
Create a list of entities with a given KIND that are found within another list.	MALKL

4.4.3 GENERAL OPERATIONS

The following table presents the general routines:

DESCRIPTION	ROUTINE
Creates an empty list.	MAL
Create an empty list of specified size.	MALN
Makes a copy of a list.	MALCPY
Find the position of an entity in a list.	MALFND
Count the entities in a list.	MALNO
Get the Nth key from a list.	MALGTK

4.4.4 DELETE OPERATIONS

The following table presents the delete routines:

DESCRIPTION	ROUTINE
Delete an application list.	MALD
Delete all application lists.	MALDA
Delete an application list and all lists created after it.	MALDI
Set or unset the application list lock flag.	MALOCK

4.4.5 EDIT OPERATIONS

The following table presents the edit routines:

DESCRIPTION	ROUTINE
Attach an entity or list of entities to a list.	MALATC
Insert an entity or list of entities into a list.	MALINS
Remove duplicate entries in a list.	MALRDE
Replace a list.	MALREP
Remove an entity from a list.	MALRMV
Reorder list of entities in user to constituent order.	MALROR
Replace an entity in a list.	MALRPL
Reverse the order of a list	MALRVS
Sorts a list using order provided by a user defined function	MALSRT

4.4.6 SEQUENTIAL READ AND EXECUTE OPERATIONS

The following table shows routines that process a list sequentially (as if it were a file):

DESCRIPTION	ROUTINE
Read the next entry in a list.	MALRD
Setup for reading in a forward direction.	MALSTF
Setup for reading in reverse direction.	MALSTR
Execute a procedure on an entity or a list of entities.	MAEXEQ
Execute a procedure on all entities of a specified KIND.	MAKXEQ
Execute a procedure on an entity or a list of entities.	MALXEQ
Execute a given procedure on constituents of entity.	MAECXQ
Execute a procedure on the users of an entity.	MAEUXQ

The MALSTF and MALSTR set up a list for forward or reverse reading of an application list. Forward reading is assumed and need not be called explicitly before a read or an execute function is used. However, after an end-of-list is signaled, the list is disabled. An explicit setup must be done to enable the list.

SECTION 5

GENERAL UTILITIES

This section contains descriptions of available general utility routines, as shown in the table below.

DESCRIPTION	ROUTINE
Get number of different KIND values in the working-form model.	MAECTK
Get KIND value stored at specific position in KIND list.	MAEKND
Determine if an entity has any users.	MAEUSR
Get actual model space used and amount of model free space.	MASMSZ
Determine the number of entities in the model of a specified KIND.	MAKCNT

APPENDICES

INTRODUCTION

Valuable supplementary information not included earlier in this manual is included in this section.

Page

A ACCESS SOFTWARE CALLING PARAMETER INDEX. A-1

B ALPHABETICAL ACCESS SOFTWARE ROUTINE INDEX B-1

C ACCESS SOFTWARE RETURN CODE INDEX. C-1

D ACCESS SOFTWARE FORTRAN SCHEMA DIAGRAM D-1

E ACCESS SOFTWARE PASCAL SCHEMA DIAGRAM. E-1

F GENERAL TECHNIQUES/GUIDELINES. F-1

G RUN-TIME ENVIRONMENT G-1
    INTRODUCTION. G-1
    INTERLANGUAGE CONVENTIONS G-2
    ESTABLISHING INTERLANGUAGE ENVIRONMENT. G-3
    REGISTER CONVENTIONS. G-5
    PASCAL DYNAMIC STORAGE AREA G-6
    EXAMPLES. G-8

H ERROR AND WARNING RETURN CODE INDEX. H-1

SECTION 6

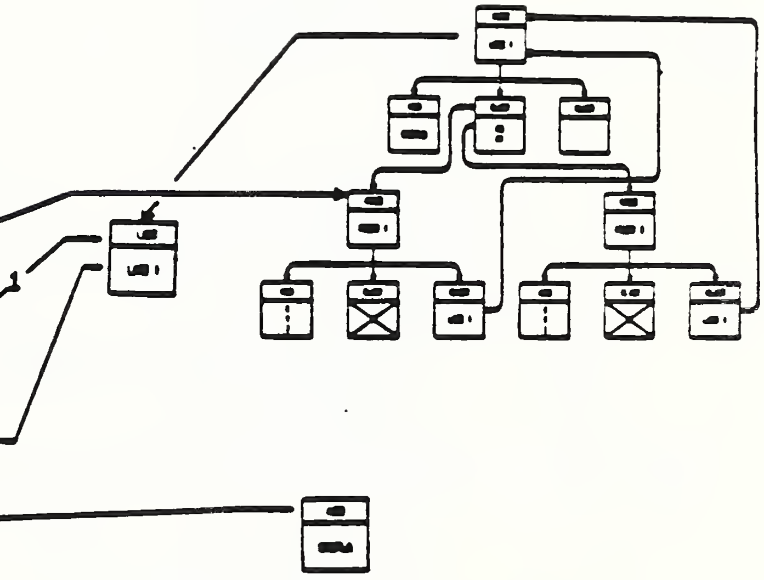
SAMPLE PROGRAMS

The following pages illustrate uses of the Access Software. These example
show Create and Get operations for a line.

DESCRIPTION	MAS ROUTINES USED
Create a Line	MAL, MALATC, MAECR, MALD
Get Constituents	MALNO, MALGTK
Get Users	MAEU, MALNO, MALGTK

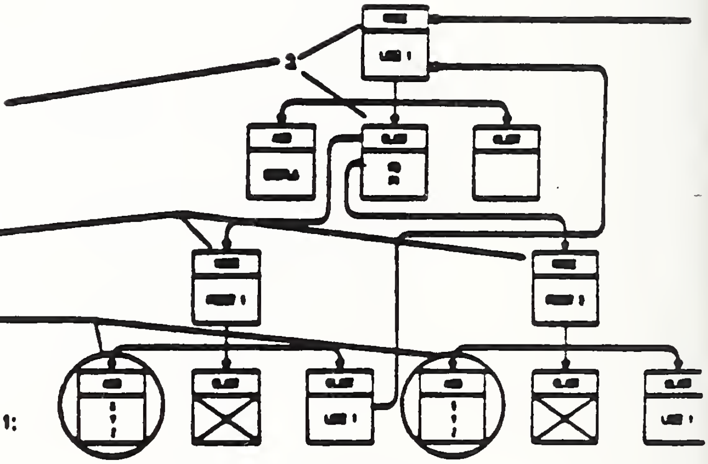
Line
Get Users

```
PROCEDURE USERS  :
VAR
  BUFFER         : ENTBLOCK ;
  RC             : EXT_RET_CODE ;
  KEYE_SNB       : ENTKEY ;
  KEYLC          : LISTKEY ;
  NEXT_USER      : INTEGER ;
  NUM_OF_USERS   : INTEGER ;
BEGIN
   .
   .  SELECT ENTITY
   .
(*  GET USER LIST *)
    MAGU (KEYE_SNB,KEYLC,RC) ;

(*  GET NUMBER IN LIST *)
    MALNO (KEYLC,NUM_OF_USERS,RC) ;

    WHILE NUM_OF_USERS > 0 DO BEGIN
      (* GET ENTITY FROM LIST*)
      MALSTE (KEYLC,NEXT_USER, KEYE_SNB,RC) ;

      (* GET ABS FROM ENTITY*)
      MAGSTE (KEYE_SNB,BUFFER,RC) ;
      .
      .  DISPLAY ENTITY INFORMATION
      .
      NEXT_USER := NEXT_USER + 1 ;
      NUM_OF_USERS := NUM_OF_USERS - 1 ;
      END ;
END ;
```



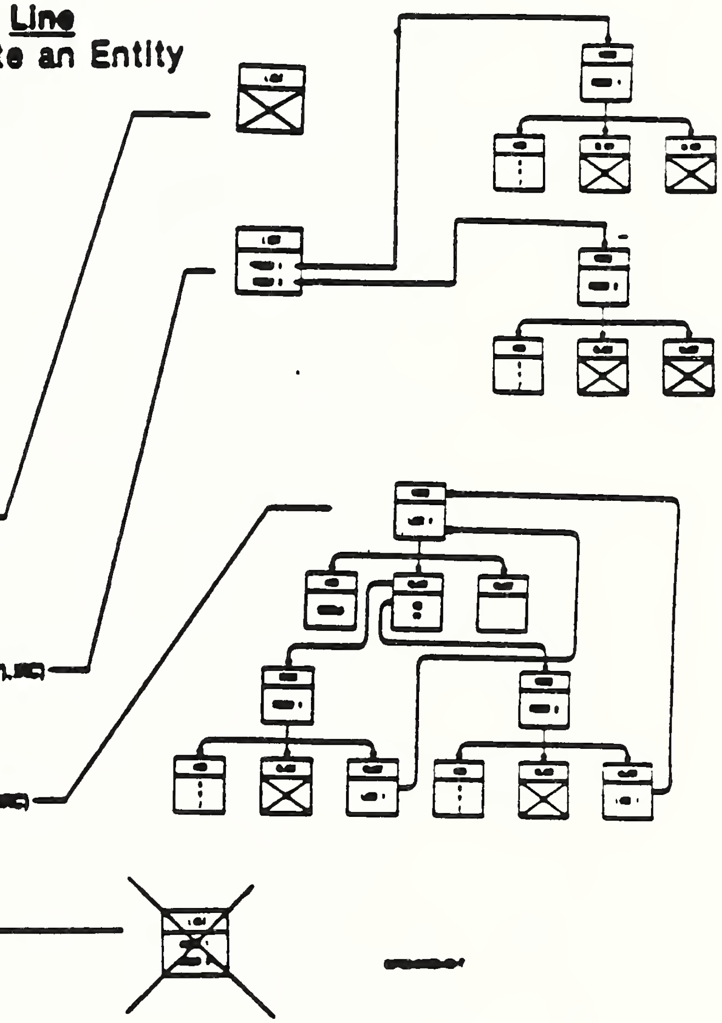
Line
Get Constituents

```
  EXAMP DATA/CNSTNTS)
PROCEDURE CNSTNTS;
VAR
  BUFFER          : ENTBLOCK ;
  RC              : EXT_RET_CODE ;
  CNSTNT_KEY      : ENTKEY ;
  KEYLC           : LISTKEY ;
  NEXT_CNSTNT_NO : INTEGER ;
  NUM_OF_CONSTITUENTS : INTEGER ;
  NUM_OF_DATA_ENT : INTEGER ;
  RESPONSE        : ORRESPONSE ;
BEGIN
      .
  VERIFY INPUT AND INITIALIZE
      .
     NEXT_CNSTNT_NO := 1 ;
(*   DETERMINE NUMBER OF CONSTITS *)
     MALNO (INPUTKEY,NUM_OF_CONSTITUENTS, RC) ;

       WHILE NUM_OF_CONSTITUENTS > 0
       DO BEGIN
(*       GET ENTITY KEY FROM LIST *)
         MALSTK (INPUTKEY,NEXT_CNSTNT_NO,
                CNSTNT_KEY,RC) ;
(*       GET ENTITY ADB *)
         MAGENT (CNSTNT_KEY,BUFFER,RC) ;
         .
         USE THE DATA IN THE ADB
         .
         NEXT_CNSTNT_NO := NEXT_CNSTNT_NO + 1;
         NUM_OF_CONSTITUENTS :=
               NUM_OF_CONSTITUENTS - 1 ;
       END ;
END ;
```



Line
Create an Entity

```
EXAMP.DATA(CRETLN)
CC
          KIND    = KLINE
          LENGTH  = LLINE
          SYSUSE  = 0
          VERSUN  = 0
          SYSID   = KIDENT
          IDENT   = JDENT
          DISPLA  = DISPLY
          RGB(1)  = 0
          RGB(2)  = 0
          RGB(3)  = 0
          NITEMS  = 0
          SYMBOL  = DISTYP
C
C            CREATE AN EMPTY LIST
C
          CALL MAL (LIST,IRC)
          IF(IRC .NE. 0) GOTO 900
C
C    CREATE CONSTITUENT LIST OF LINE
C
            DO 100 I = 1,NLINE
            CALL MALATC(LIST,KEYPNT(I),IRC)
   100    CONTINUE
C
C            CREATE A LINE ENTITY
C
          CALL MAECR(KGL, LIST, KEY, IRC)
C
          IF(IRC .NE. 0) GOTO 900
C
C    DELETE LIST
          CALL MALD(LIST,IRC)
C
```



5.0 STEP DATABASE IMPLEMENTATIONS

This chapter will contain descriptions of a variety of STEP Database implementations. There is no information for this section at this time.

6.0 STEP SOFTWARE BINDINGS

There is no information for this section at this time.

CONTENTS

7.1 Notational Conventions 2
7.2 Predefined Functions 3
7.3 CARTESIAN_THREE_COORDINATE 4
7.3.1 From a 3D Cartesian Point 4
7.3.2 To a 3D Cartesian Point 4
7.4 LINE_SEGMENT 5
7.4.1 To a 3D Bounded Line 5
7.4.2 From a 3D Bounded Line 5
7.5 CIRCLE 6
7.5.1 To a 3D 2-Point & Normal vector defined Circle . . 6
7.5.2 From a 3D 2-Point & Normal vector defined Circle . 6
7.6 ELLIPSE 7
7.6.1 From a 3 Point Ellipse 7
7.7 TRIMMED CURVE 8
7.7.1 To a 3D 3-Point Circular Arc 8
7.7.2 From a 3D 3-Point Circular Arc 9
7.8 Parabolic Arc 12
7.8.1 To a 5 point conic 12
7.9 BEZIER_CURVE 15
7.9.1 To a Single Segment Hermitian Parametric Cubic Curve 15
7.9.2 From a Single Segment Hermitian Parametric Cubic Curve 16
7.10 SPHERICAL_SURFACE 17
7.10.1 To a Center point & vector with magnitude defined Sphere 17
7.10.2 From a Center point & vector with magnitude defined Sphere 17
7.11 PLANE 18
7.11.1 To a Point and Vector defined Plane 18
7.11.2 From a Point and Vector defined Plane 18
7.12 BEZIER SURFACE 18
7.12.1 From a Hermitian surface to a Bezier surface . . 18
7.12.2 From a Bezier surface to a Hermitian surface . . 2[illegible]
7.13 Conversion Issues 29

GEOMETRIC CONVERSION ALGORITHMS

This chapter provides conversion algorithms which may be used to convert the STEP geometric entities to and from other geometric forms. This chapter of Annex D is organized by the STEP entity type. Under each entity type will be specific types of conversion algorithms. The conversion algorithms presented here are volunteered from a variety of sources. The reader is cautioned to read each algorithm carefully, checking for possible errors or efficiency improvements. These algorithms are expected to improve over time. The reader should also note that navigation within local data structures is not accounted for in these algorithms. An algorithm would assume for example, that the value Point.X, is somehow made available from the system in order to take part in calculations ; the details of how it is made available are not of concern to the algorithm. Some entity types may have more types of conversion algorithms than others.

7.1 Notational Conventions

In this document the following symbols and keywords are used to indicate various functions or actions.

SYMBOL	MEANING
:=	Assignment
:==	Structure assignment { all named components of the source data structure } { are assigned to like named components of the } { target data structure. }
-	Scalar subtraction
+	Scalar addition
*	Scalar multiplication
/	Scalar division
·	Matrix dot product
X	Matrix cross product

KEYWORD	MEANING
create	The entity definition is complete.
return	Designates the value a function will return.

7.2 Predefined Functions

The following functions will be referenced by the various conversion algorithms :

```
function MAGNITUDE( A: vector );
  {  This function will determine the  }
  {  magnitude of a vector.            }

  temp_value := SQRT( A.I **2 + A.J **2 + A.K **2 );
  return temp_value
end function;


function NORMALIZE( A: vector );
  {  This function will normalize a     }
  {  vector by dividing each component  }
  {  of the vector by the vector's      }
  {  magnitude.                         }

  temp_value := magnitude( A );
  temp_vector.I := A.I / temp_value;
  temp_vector.J := A.J / temp_value;
  temp_vector.K := A.K / temp_value;
  return temp_vector
end function;


function VECTOR( A: location, B: location );
  {  This function, given two points,  }
  {  will calculate the vector from    }
  {  point A to point B.  The vector   }
  {  is not normalized.                }

  temp_vector.i := B.x - A.x;
  temp_vector.j := B.y - A.y;
  temp_vector.k := B.z - A.z;
  return temp_vector
end function;
```

7.3 CARTESIAN_THREE_COORDINATE

7.3.1 From a 3D Cartesian Point

```
{  The 3D Cartesian POINT entity will be translated into the    }
{  following entities:                                          }
{    1  CARTESIAN_THREE_COORDINATE                              }

CARTESIAN_THREE_COORDINATE.X := POINT.X;
CARTESIAN_THREE_COORDINATE.Y := POINT.Y;
CARTESIAN_THREE_COORDINATE.Z := POINT.Z;
create CARTESIAN_THREE_COORDINATE
```

7.3.2 To a 3D Cartesian Point

```
{  The STEP CARTESIAN_THREE_COORDINATE entity will be trans-    }
{  lated into the following entities:                           }
{    1  3D POINT                                                }

POINT.X := CARTESIAN_THREE_COORDINATE.X;
POINT.Y := CARTESIAN_THREE_COORDINATE.Y;
POINT.Z := CARTESIAN_THREE_COORDINATE.Z;
create POINT
```

7.4 LINE_SEGMENT

7.4.1 To a 3D Bounded Line

```
{  The STEP LINE_SEGMENT entity will be translated into the
{  following  entities:
{    1  LINE


LINE.P0 :== LINE_SEGMENT.FIRST_POINT->CARTESIAN_THREE_COORDINAT
LINE.P1 :== LINE_SEGMENT.LAST_POINT->CARTESIAN_THREE_COORDINATE
create LINE
```

7.4.2 From a 3D Bounded Line

```
{  The LINE entity will be translated into the following
{  entities:
{    2  CARTESIAN_THREE_COORDINATE (possibly embedded/referred)
{    1  LINE_SEGMENT

LINE_SEGMENT.FIRST_POINT := LINE.P0;
LINE_SEGMENT.LAST_POINT  := LINE.P1;
create LINE_SEGMENT
```

7.5 CIRCLE

7.5.1 To a 3D 2-Point & Normal vector defined Circle

```
{  The STEP CIRCLE entity will be translated into the following }
{  entities:                                                    }
{    1  CIRCLE                                                  }

CIRCLE.PC   :== CIRCLE.POSITION->AXIS2_PLACEMENT.LOCATION;
CIRCLE.AXIS :== CIRCLE.POSITION->AXIS2_PLACEMENT.AXIS;

temp_vector :== CIRCLE.POSITION->AXIS2_PLACEMENT.REF_DIRECTION;
temp_vector :== normalize( temp_vector );

CIRCLE.P0.X := CIRCLE.POSITION->AXIS2_PLACEMENT.LOCATION.X +
                                  temp_vector.I * CIRCLE.RADIUS;
CIRCLE.P0.Y := CIRCLE.POSITION->AXIS2_PLACEMENT.LOCATION.Y +
                                  temp_vector.J * CIRCLE.RADIUS;
CIRCLE.P0.Z := CIRCLE.POSITION->AXIS2_PLACEMENT.LOCATION.Z +
                                  temp_vector.K * CIRCLE.RADIUS;
create CIRCLE
```

7.5.2 From a 3D 2-Point & Normal vector defined Circle

```
{  The  3 Point CIRCLE entity will be translated into the       }
{  following entities:                                          }
{    1 CIRCLE                                                   }
{    1 CARTESIAN_THREE_COORDINATE                               }
{    1 AXIS2_PLACEMENT                                          }
{    2 THREE_SPACE_DIRECTION                                    }

CARTESIAN_THREE_COORDINATE :== CIRCLE.PC;
create CARTESIAN_THREE_COORDINATE;

THREE_SPACE_DIRECTION(1).X := CIRCLE.AXIS.I;
THREE_SPACE_DIRECTION(1).Y := CIRCLE.AXIS.J;
THREE_SPACE_DIRECTION(1).Z := CIRCLE.AXIS.K;
create THREE_SPACE_DIRECTION(1);
THREE_SPACE_DIRECTION(2) :== vector( CIRCLE.PC, CIRCLE.P0 );
create THREE_SPACE_DIRECTION(2);

AXIS2_PLACEMENT.LOCATION      := CARTESIAN_THREE_COORDINATE;
AXIS2_PLACEMENT.AXIS          := THREE_SPACE_DIRECTION(1);
AXIS2_PLACEMENT.REF_DIRECTION := THREE_SPACE_DIRECTION(2);
create AXIS2_PLACEMENT;

CIRCLE.RADIUS := magnitude( THREE_SPACE_DIRECTION(2) );
CIRCLE.POSITION := AXIS2_PLACEMENT;
create CIRCLE
```

7.6 ELLIPSE

7.6.1 From a 3 Point Ellipse

```
{  The ELLIPSE entity will be translated into the following
{  entities:
{     1  ELLIPSE
{     1  AXIS2_PLACEMENT
{     1  CARTESIAN_THREE_COORDINATE
{     2  THREE_SPACE_DIRECTION

temp_vector.i := ELLIPSE.MAJOR.X - ELLIPSE.PC.X;
temp_vector.j := ELLIPSE.MAJOR.Y - ELLIPSE.PC.Y;
temp_vector.k := ELLIPSE.MAJOR.Z - ELLIPSE.PC.Z;

THREE_SPACE_DIRECTION(1).i := ELLIPSE.MINOR.X - ELLIPSE.PC.X;
THREE_SPACE_DIRECTION(1).j := ELLIPSE.MINOR.Y - ELLIPSE.PC.Y;
THREE_SPACE_DIRECTION(1).k := ELLIPSE.MINOR.Z - ELLIPSE.PC.Z;

temp_magnitude := magnitude( THREE_SPACE_DIRECTION );
THREE_SPACE_DIRECTION(1) :== normalize(THREE_SPACE_DIRECTION(1))
create( THREE_SPACE_DIRECTION(1) );


THREE_SPACE_DIRECTION(2) := THREE_SPACE_DIRECTION(1) X
temp_vector;
THREE_SPACE_DIRECTION(2) :== normalize(THREE_SPACE_DIRECTION(2))
create( THREE_SPACE_DIRECTION(2) );

CARTESIAN_THREE_COORDINATE :== ELLIPSE.PC;
create( CARTESIAN_THREE_COORDINATE );

AXIS2_PLACEMENT.LOCATION      := CARTESIAN_THREE_COORDINATE;
AXIS2_PLACEMENT.AXIS          := THREE_SPACE_DIRECTION(1);
AXIS2_PLACEMENT.REF_DIRECTION := THREE_SPACE_DIRECTION(1);
create( AXIS2_PLACEMENT );

ELLIPSE.SEMI_AXIS_1 := magnitude( temp_vector );
ELLIPSE.SEMI_AXIS_2 := temp_magitude;
ELLIPSE.POSITION    := AXIS2_PLACEMENT;
create( ELLIPSE )
```

7.7 TRIMMED CURVE

7.7.1 To a 3D 3-Point Circular Arc

```
if TRIMMED_CURVE uses( CIRCLE ) then do

  for each TRIMMED_CURVE in users( CIRCLE ) do

    ARC.P0 :== TRIMMED_CURVE.POINT_1->CARTESIAN_THREE_COORDINATE;
    ARC.P1 :== TRIMMED_CURVE.POINT_2->CARTESIAN_THREE_COORDINATE;

    temp_center.X := CIRCLE.POSTITION->AXIS2_PLACEMENT.LOCATION->
                     CARTESIAN_THREE_COORDINATE.X;
    temp_center.Y := CIRCLE.POSTITION->AXIS2_PLACEMENT.LOCATION->
                     CARTESIAN_THREE_COORDINATE.Y;
    temp_center.Z := CIRCLE.POSTITION->AXIS2_PLACEMENT.LOCATION->
                     CARTESIAN_THREE_COORDINATE.Z;

    temp_vector.I := ( ARC.P0.X + ARC.P1.X )/2 - temp_center.X;
    temp_vector.J := ( ARC.P0.Y + ARC.P1.Y )/2 - temp_center.Y;
    temp_vector.K := ( ARC.P0.Z + ARC.P1.Z )/2 - temp_center.Z;

    temp_vector :== normalize( temp_vector );

    ARC.PM.X := temp_center.X + temp_vector.I * CIRCLE.RADIUS;
    ARC.PM.Y := temp_center.Y + temp_vector.J * CIRCLE.RADIUS;
    ARC.PM.Z := temp_center.Z + temp_vector.K * CIRCLE.RADIUS;

    create ARC
  end
end
```

7.7.2 From a 3D 3-Point Circular Arc

```
{  The  ARC entity will be translated into the following          }
{  entities:                                                      }
{    1 TRIMMED_CURVE                                              }
{    1 CIRCLE
{    3 CARTESIAN_THREE_COORDINATE                                 }
{    1 AXIS2_PLACEMENT                                            }
{    2 THREE_SPACE_DIRECTION                                      }


{                                                                 }
{  Construct vectors from PM to P0 and P1                         }
{                                                                 }
temp_vector(1) :== vector( ARC.PM, ARC.P0 );
temp_vector(2) :== vector( ARC.PM, ARC.P1 );


{                                   --->   --->
{  Calculate the cross product:  PMP1 X PMP0  to produce a
{  vector normal to the plane of the CIRCLE.
{
    temp_vector(3) :== temp_vector(2) X temp_vector(1);

{
{  Normalize the normal vector
{
THREE_SPACE_DIRECTION(1) :== normalize( temp_vector(3) );
create THREE_SPACE_DIRECTION(1);

{
{  Find the coordinates of the midpoints of the line segments:
{  PMP0 and PMP1.
{
    temp_point(1) :== (1/2)*(ARC.PM + ARC.P0);
    temp_point(2) :== (1/2)*(ARC.PM + ARC.P1);

{
{  The equation of the plane normal to the vector PMP0 is:
{
{    X * temp_vector(1).I +
{    Y * temp_vector(1).J +
{    Z * temp_vector(1).K  = CONSTANT
{  The value of CONSTANT is the equation evaluated at some
{  point on the plane.  e.g. temp_point(1)
{
{  similarly the constant of the second plane in found by
{  evaluating the equation at temp_point(2).
{
    temp_const(1) := temp_point(1).X * temp_vector(1).I +
                     temp_point(1).Y * temp_vector(1).J +
                     temp_point(1).Z * temp_vector(1).K;
    temp_const(2) := temp_point(2).X * temp_vector(2).I +
                     temp_point(2).Y * temp_vector(2).J +
                     temp_point(2).Z * temp_vector(2).K;
```

```
{                                                              }
{  The equation of the plane that contains the circle can be   }
{  found by a similar manner.                                  }
{                                                              }
    temp_const(3) := ARC.PM.X * temp_vector(3).I +
                     ARC.PM.Y * temp_vector(3).J +
                     ARC.PM.Z * temp_vector(3).K;

{                                                              }
{  Both of the planes must contain the center point of the     }
{  circle because of basic geometry rules.  Also the plane on  }
{  which the circular_arc lies must contain the center point of }
{  the base circle.  Now solve the three equations for three   }
{  unknowns (the coordinates of the center points) using       }
{  Gaussian elimination and backwards substitution on the      }
{  matrix:                                                     }
{                                                              }
{    +-                                                 -+     }
{    |   tv(1).I   tv(1).J   tv(1).K   |   temp_const(1)   |   }
{    |                                 |                   |   }
{    |   tv(1).I   tv(1).J   tv(1).K   |   temp_const(2)   |   }
{    |                                 |                   |   }
{    |   tv(1).I   tv(1).J   tv(1).K   |   temp_const(3)   |   }
{    +-                                                 -+     }
{          *** 'tv' denotes 'temp_vector' ***                  }
{                                                              }
{  The results are stored as:  X in temp_const(1)              }
{                              Y in temp_const(2)              }
{                              Z in temp_const(3)              }
{                                                              }


perform GAUSSIAN_ELEMINATION;

CARTESIAN_THREE_COORDINATE(1).X := temp_const(1);
CARTESIAN_THREE_COORDINATE(1).Y := temp_const(2);
CARTESIAN_THREE_COORDINATE(1).Z := temp_const(3);
create CARTESIAN_THREE_COORDINATE(1);


{                                                              }
{  Create the reference vector relative to the ARC.PM coordi-  }
{  nate.                                                       }
{                                                              }
THREE_SPACE_DIRECTION(2) :==
                 vector( CARTESIAN_THREE_COORDINATE(1),ARC.P0 );
create THREE_SPACE_DIRECTION(2);

AXIS2_PLACEMENT.LOCATION      := CARTESIAN_THREE_COORDINATE(1);
AXIS2_PLACEMENT.AXIS          := THREE_SPACE_DIRECTION(1);
AXIS2_PLACEMENT.REF_DIRECTION := THREE_SPACE_DIRECTION(2);
create AXIS2_PLACEMENT;

CIRCLE.RADIUS := magnitude( THREE_SPACE_DIRECTION(2) );
CIRCLE.POSITION := AXIS2_PLACEMENT;
```

```
create CIRCLE;

{
{  Find the endpoints of the circular_arc
{
CARTESIAN_THREE_COORDINATE(2) :== ARC.P0;
create CARTESIAN_THREE_COORDINATE(2);

CARTESIAN_THREE_COORDINATE(3) :== ARC.P1;
create CARTESIAN_THREE_COORDINATE(3);

{
{  Use the endpoint version (not the parametric version) to
{  define a TRIMMED_CURVE.
{
TRIMMED_CURVE.BASE_CURVE  := CIRCLE;
TRIMMED_CURVE.PARAMETER_1 := 0;
TRIMMED_CURVE.PARAMETER_2 := 0;
TRIMMED_CURVE.POINT_1     := CARTESIAN_THREE_COORDINATE(2);
TRIMMED_CURVE.POINT_2     := CARTESIAN_THREE_COORDINATE(3);
TRIMMED_CURVE.SENSE       := TRUE;
create TRIMMED_CURVE
```

7.8 Parabolic Arc

7.8.1 To a 5 point conic

```
{  The STEP PARABOLA will be translated into the following       }
{  entities:                                                      }
{    1  CONIC                                                     }

if TRIMMED_CURVE in users( PARABOLA ) do
  for each TRIMMED_CURVE in users( PARABOLA ) do begin

    z_vector.i := PARABOLA.POSITION->AXIS2_PLACEMENT.AXIS ->
                                     THREE_SPACE_DIRECTION.X;
    z_vector.j := PARABOLA.POSITION->AXIS2_PLACEMENT.AXIS ->
                                     THREE_SPACE_DIRECTION.Y;
    z_vector.k := PARABOLA.POSITION->AXIS2_PLACEMENT.AXIS ->
                                     THREE_SPACE_DIRECTION.Z;
    z_vector :== normalize( z_vector );

    trans[3,1] := z_vector.i;
    trans[3,1] := z_vector.j;
    trans[3,1] := z_vector.k;

    temp_vector.i := PARABOLA.POSITION->
          AXIS2_PLACEMENT.REF_DIRECTION->THREE_SPACE_DIRECTION.X;
    temp_vector.j := PARABOLA.POSITION->
          AXIS2_PLACEMENT.REF_DIRECTION->THREE_SPACE_DIRECTION.Y;
    temp_vector.k := PARABOLA.POSITION->
          AXIS2_PLACEMENT.REF_DIRECTION->THREE_SPACE_DIRECTION.Z;

    constant := temp_vector · z_vector;
    x_vector.i := temp_vector.i - constant*z_vector.i;
    x_vector.j := temp_vector.j - constant*z_vector.j;
    x_vector.k := temp_vector.k - constant*z_vector.k;
    x_vector :== normalize( x_vector );

    trans[1,1] := x_vector.i;
    trans[1,2] := x_vector.j;
    trans[1,3] := x_vector.k;

    y_vector := z_vector X x_vector;
    y_vector := normalize( y_vector );

    trans[2,1] := y_vector.i;
    trans[2,2] := y_vector.j;
    trans[2,3] := y_vector.k;

    temp_p0.x := TRIMMED_CURVE.POINT_1->
          CARTESIAN_THREE_COORDINATE.X - PARABOLA.POSITION->
          AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.X;
    temp_p0.y := TRIMMED_CURVE.POINT_1->
          CARTESIAN_THREE_COORDINATE.Y - PARABOLA.POSITION->
          AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.Y;
    temp_p0.z := TRIMMED_CURVE.POINT_1->
          CARTESIAN_THREE_COORDINATE.Z - PARABOLA.POSITION->
    AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.

temp_p4.x := TRIMMED_CURVE.POINT_2->
    CARTESIAN_THREE_COORDINATE.X - PARABOLA.POSITION->
    AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.
temp_p4.y := TRIMMED_CURVE.POINT_2->
    CARTESIAN_THREE_COORDINATE.Y - PARABOLA.POSITION->
    AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.
temp_p4.z := TRIMMED_CURVE.POINT_2->
    CARTESIAN_THREE_COORDINATE.Z - PARABOLA.POSITION->
    AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.

local_p0 :== trans * temp_p0;
local_p4 :== trans * temp_p4;

t0 := ( temp_p0.y - PARABOLA.POSITION->
  AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.Y )
  /(2*PARABOLA.FOCAL_DIST);
t1 := ( temp_p1.y - PARABOLA.POSITION->
  AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE.Y )
  /(2*PARABOLA.FOCAL_DIST);


local_p1.x := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.X + PARABOLA.FOCAL_DIST *
  ( t0*(3/4) + t1*(1/4) )**2;
local_p1.y := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.Y + PARABOLA.FOCAL_DIST *
  ( t0*(3/4) + t1*(1/4) )*2;
local_p1.z := 0;

local_p2.x := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.X + PARABOLA.FOCAL_DIST *
  ( t0*(2/4) + t1*(2/4) )**2;
local_p2.y := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.Y + PARABOLA.FOCAL_DIST *
  ( t0*(2/4) + t1*(2/4) )*2;
local_p2.z := 0;

local_p3.x := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.X + PARABOLA.FOCAL_DIST *
  ( t0*(1/4) + t1*(3/4) )**2;
local_p3.y := PARABOLA.POSITION->AXIS2_PLACEMENT.LOCATION->
  CARTESIAN_THREE_COODINATE.Y + PARABOLA.FOCAL_DIST *
  ( t0*(1/4) + t1*(3/4) )*2;
local_p3.z := 0;

CONIC.P0 := trans -¹ * local_p0 + PARABOLA.POSITION->
     AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE
CONIC.PA := trans -¹ * local_p1 + PARABOLA.POSITION->
     AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE
CONIC.PB := trans -¹ * local_p2 + PARABOLA.POSITION->
     AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE
CONIC.PC := trans -¹ * local_p3 + PARABOLA.POSITION->
     AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE
    CONIC.P1 := trans -1 * local_p4 + PARABOLA.POSITION->
            AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINATE;

    create CONIC
  end
end;
```

7.9 BEZIER_CURVE

7.9.1 To a Single Segment Hermitian Parametric Cubic Curve

```
{  The STEP BEZIER CURVE will be translated into the following
{  entities:
{     1  PARAMETRIC CUBIC CURVE

if( BEZIER_CURVE.RATIONAL ) then do

  TBD

end
else do

  CUBIC.P0 :== BEZIER_CURVE.CONTROL_POINTS[1]->
                                              CARTESIAN_THREE_COORDINAT

  CUBIC.V0.I := 3*( BEZIER_CURVE.CONTROL_POINTS[2].X -
                    BEZIER_CURVE.CONTROL_POINTS[1].X );
  CUBIC.V0.J := 3*( BEZIER_CURVE.CONTROL_POINTS[2].Y -
                    BEZIER_CURVE.CONTROL_POINTS[1].Y );
  CUBIC.V0.K := 3*( BEZIER_CURVE.CONTROL_POINTS[2].Z -
                    BEZIER_CURVE.CONTROL_POINTS[1].Z );
  CUBIC.V1.I := 3*( BEZIER_CURVE.CONTROL_POINTS[4].X -
                    BEZIER_CURVE.CONTROL_POINTS[3].X );
  CUBIC.V1.J := 3*( BEZIER_CURVE.CONTROL_POINTS[4].Y -
                    BEZIER_CURVE.CONTROL_POINTS[3].Y );
  CUBIC.V1.K := 3*( BEZIER_CURVE.CONTROL_POINTS[4].Z -
                    BEZIER_CURVE.CONTROL_POINTS[3].Z );

  CUBIC.P0 :== BEZIER_CURVE.CONTROL_POINTS[2]->
                                              CARTESIAN_THREE_COORDINAT

  create CUBIC
end
```

7.9.2 From a Single Segment Hermitian Parametric Cubic Curve

```
{  The PARAMETRIC CUBIC entity will be translated into the           }
{  following entities:                                               }
{    1  BEZIER_CURVE                                                 }
{    4  CARTESIAN_THREE_COORDINATE                                   }

CARTESIAN_THREE_COORDINATE(1) :== CUBIC.P0;
create CARTESIAN_THREE_COORDINATE(1);

CARTESIAN_THREE_COORDINATE(2).X := CUBIC.P0.X + (1/3)*CUBIC.V0.I;
CARTESIAN_THREE_COORDINATE(2).Y := CUBIC.P0.Y + (1/3)*CUBIC.V0.J;
CARTESIAN_THREE_COORDINATE(2).Z := CUBIC.P0.Z + (1/3)*CUBIC.V0.K;
create CARTESIAN_THREE_COORDINATE(2);

CARTESIAN_THREE_COORDINATE(3).X := CUBIC.P1.X - (1/3)*CUBIC.V1.I;
CARTESIAN_THREE_COORDINATE(3).Y := CUBIC.P1.Y - (1/3)*CUBIC.V1.J;
CARTESIAN_THREE_COORDINATE(3).Z := CUBIC.P1.Z - (1/3)*CUBIC.V0.K;
create CARTESIAN_THREE_COORDINATE(3);

CARTESIAN_THREE_COORDINATE(4) :== CUBIC.P1;
create CARTESIAN_THREE_COORDINATE(4);

BEZIER_CURVE.RATIONAL := FALSE;
BEZIER_CURVE.DEGREE := 3;
BEZIER_CURVE.CONTROL_POINTS := ( CARTESIAN_THREE_COORDINATE(1),
  CARTESIAN_THREE_COORDINATE(2), CARTESIAN_THREE_COORDINATE(3),
  CARTESIAN_THREE_COORDINATE(4) );
BEZIER_CURVE.WEIGHTS := 0;
BEZIER_CURVE.FORM_NUMBER := 0;
BEZIER_CURVE.SELF_INTERSECT := UNKNOWN;
create BEZIER_CURVE
```

7.10 SPHERICAL_SURFACE

7.10.1 To a Center point & vector with magnitude defined Spher

```
{  The STEP SPHERICAL_SURFACE entity will be translated into
{  the following entities:
{    1  SPHERE

SPHERE.CENTER :== SPHERICAL_SURFACE->
            AXIS2_PLACEMENT.LOCATION->CARTESIAN_THREE_COORDINAT

SPHERE.RADVEC.I := SPHERICAL_SURFACE->AXIS2_PLACEMENT.AXIS->
                         THREE_SPACE_DIRECTION.X * SPHERE.RADIU
SPHERE.RADVEC.J := SPHERICAL_SURFACE->AXIS2_PLACEMENT.AXIS->
                         THREE_SPACE_DIRECTION.Y * SPHERE.RADIU
SPHERE.RADVEC.K := SPHERICAL_SURFACE->AXIS2_PLACEMENT.AXIS->
                         THREE_SPACE_DIRECTION.Z * SPHERE.RADIU
create SPHERE
```

7.10.2 From a Center point & vector with magnitude defined Sphere

```
{  The SPHERE entity will be translated into the following
{  entities:
{    1  SPHERICAL_SURFACE
{    1  AXIS2_PLACEMENT
{    1  CARTESIAN_THREE_COORDINATE
{    1  THREE_SPACE_DIRECTION

CARTESIAN_THREE_COORDINATE :== SPHERE.CENTER;
create CARTESIAN_THREE_COORDINATE;

THREE_SPACE_DIRECTION :== SPHERE.RADVEC;
temp_value := magnitude( THREE_SPACE_DIRECTION );
THREE_SPACE_DIRECTION :== normalize( THREE_SPACE_DIRECTION );
create THREE_SPACE_DIRECTION;

AXIS2_PLACEMENT.LOCATION      := CARTESIAN_THREE_COORDINATE;
AXIS2_PLACEMENT.AXIS          := THREE_SPACE_DIRECTION;
AXIS2_PLACEMENT.REF_DIRECTION := 0;
create AXIS2_PLACEMENT;

SPHERICAL_SURFACE.RADIUS   := temp_value;
SPHERICAL_SURFACE.POSITION := AXIS2_PLACEMENT;
create SPHERICAL_SURFACE
```

7.11 PLANE

7.11.1 To a Point and Vector defined Plane

```
{  The STEP PLANE entity will be translated into the following  }
{  entities:                                                    }
{    1  PLANE                                                   }

    PLANE.ORG :== PLANE.POSITION->AXIS2_PLACEMENT.LOCATION->
                                          CARTESIAN_THREE_COORDINATE;
    PLANE.DIR :== PLANE.POSITION->AXIS2_PLACEMENT.AXIS->
                                          THREE_SPACE_DIRECTION;
    create PLANE
```

7.11.2 From a Point and Vector defined Plane

```
{  The  PLANE entity will be translated into the following      }
{  entities:                                                    }
{    1  PLANE                                                   }
{    1  AXIS2_PLACEMENT                                         }
{    1  CARTESIAN_THREE_COORDINATE                              }
{    1  THREE_SPACE_DIRECTION                                   }

CARTESIAN_THREE_COORDINATE :== PLANE.ORG;
create CARTESIAN_THREE_COORDINATE;

THREE_SPACE_DIRECTION :== PLANE.DIR;
create THREE_SPACE_DIRECTION;

AXIS2_PLACEMENT.LOCATION      := CARTESIAN_THREE_COORDINATE;
AXIS2_PLACEMENT.AXIS          := THREE_SPACE_DIRECTION;
AXIS2_PLACEMENT.REF_DIRECTION := 0;
create AXIS2_PLACEMENT;

PLANE.POSITION := AXIS2_PLACEMENT;
create PLANE
```

7.12 BEZIER SURFACE

7.12.1 From a Hermitian surface to a Bezier surface

```
{  A Hermitian surface or patch defined by 4 Hermite cubics (of }
{  degree 3) and 4 vectors is translated into the following     }
{  PDES entities:                                               }
{    1  BEZIER_SURFACE                                          }
{    16 CARTESIAN_THREE_COORDINATEs                             }


{                                                               }
{  First we must construct the geometry matrix for the patch    }
{  like this for the x coordinate:                              }
```

```
{
{     +-                                                            -+
{     |                                  :                            |
{     |    P(0,0)      P(0,1)            :   %__dx__%       %__dx__%
}
{     |                                  :   dv(0,0)        dv(0,1)    |
{     |                                  :                             |
{     |    P(1,0)      P(1,1)            :   %__dx__%       %__dx__%
}
{     |                                  :   dv(1,0)        dv(1,1)    |
{     |  ................................:..........................   |
{     |                                  :                             |
{     |    %__dx__%      %__dx__%          :   %__d²x__%     %__d²x__
|    }
{     |    du(0,0)     du(0,1)           :  dudv(0,0)     dudv(0,1)   |
{     |                                  :                             |
{     |    %__dx__%      %__dx__%          :   %__d²x__%     %__d²x__
|    }
{     |    du(1,0)     du(1,1)           :  dudv(1,0)     dudv(1,1)   |
{     |                                  :                             |
{     +-                                                            -+
{
{   and in a similar fashion for the y and z coordinates.
{

{
{   Create upper left corners of matricies
{

temp_matrix_x[1,1] := PATCH.U0CRV->CUBIC.P0.X;
temp_matrix_y[1,1] := PATCH.U0CRV->CUBIC.P0.Y;
temp_matrix_z[1,1] := PATCH.U0CRV->CUBIC.P0.Z;

temp_matrix_x[1,2] := PATCH.U0CRV->CUBIC.P1.X;
temp_matrix_y[1,2] := PATCH.U0CRV->CUBIC.P1.Y;
temp_matrix_z[1,2] := PATCH.U0CRV->CUBIC.P1.Z;

temp_matrix_x[2,1] := PATCH.U1CRV->CUBIC.P0.X;
temp_matrix_y[2,1] := PATCH.U1CRV->CUBIC.P0.Y;
temp_matrix_z[2,1] := PATCH.U1CRV->CUBIC.P0.Z;

temp_matrix_x[2,2] := PATCH.U1CRV->CUBIC.P1.X;
temp_matrix_y[2,2] := PATCH.U1CRV->CUBIC.P1.Y;
temp_matrix_z[2,2] := PATCH.U1CRV->CUBIC.P1.Z;


{
{   Create upper right corners of matricies
{

temp_matrix_x[1,3] := PATCH.U0CRV->CUBIC.V0.I;
temp_matrix_y[1,3] := PATCH.U0CRV->CUBIC.V0.J;
temp_matrix_z[1,3] := PATCH.U0CRV->CUBIC.V0.K;

temp_matrix_x[1,4] := PATCH.U0CRV->CUBIC.V1.I;
```

```
temp_matrix_y[1,4] := PATCH.U0CRV->CUBIC.V1.J;
temp_matrix_z[1,4] := PATCH.U0CRV->CUBIC.V1.K;

temp_matrix_x[2,3] := PATCH.U1CRV->CUBIC.V0.I;
temp_matrix_y[2,3] := PATCH.U1CRV->CUBIC.V0.J;
temp_matrix_z[2,3] := PATCH.U1CRV->CUBIC.V0.K;

temp_matrix_x[2,4] := PATCH.U1CRV->CUBIC.V1.I;
temp_matrix_y[2,4] := PATCH.U1CRV->CUBIC.V1.J;
temp_matrix_z[2,4] := PATCH.U1CRV->CUBIC.V1.K;

{                                                           }
{  Create lower left corners of matricies                   }
{                                                           }

temp_matrix_x[3,1] := PATCH.V0CRV->CUBIC.V0.I;
temp_matrix_y[3,1] := PATCH.V0CRV->CUBIC.V0.J;
temp_matrix_z[3,1] := PATCH.V0CRV->CUBIC.V0.K;

temp_matrix_x[3,2] := PATCH.V0CRV->CUBIC.V1.I;
temp_matrix_y[3,2] := PATCH.V0CRV->CUBIC.V1.J;
temp_matrix_z[3,2] := PATCH.V0CRV->CUBIC.V1.K;

temp_matrix_x[4,1] := PATCH.V1CRV->CUBIC.V0.I;
temp_matrix_y[4,1] := PATCH.V1CRV->CUBIC.V0.J;
temp_matrix_z[4,1] := PATCH.V1CRV->CUBIC.V0.K;

temp_matrix_x[4,2] := PATCH.V1CRV->CUBIC.V1.I;
temp_matrix_y[4,2] := PATCH.V1CRV->CUBIC.V1.J;
temp_matrix_z[4,2] := PATCH.V1CRV->CUBIC.V1.K;

{                                                           }
{  Create lower right corners of matricies                  }
{                                                           }

temp_matrix_x[3,3] := PATCH.NU0V0.I;
temp_matrix_y[3,3] := PATCH.NU0V0.J;
temp_matrix_k[3,3] := PATCH.NU0V0.K;

temp_matrix_x[3,4] := PATCH.NU0V1.I;
temp_matrix_y[3,4] := PATCH.NU0V1.J;
temp_matrix_k[3,4] := PATCH.NU0V1.K;

temp_matrix_x[4,3] := PATCH.NU1V0.I;
temp_matrix_y[4,3] := PATCH.NU1V0.J;
temp_matrix_k[4,3] := PATCH.NU1V0.K;

temp_matrix_x[4,4] := PATCH.NU1V1.I;
temp_matrix_y[4,4] := PATCH.NU1V1.J;
temp_matrix_k[4,4] := PATCH.NU1V1.K;

{                                                           }
{  Since the definition of the surface of a Hermite cubic   }
{  is:                                                      }
{    P(u,v) := U * Mh * Qh * (Mh)t * (V)t    0 <= u,v <= 1  }
```

```
{
{  Where   U      is the matrix:
{                     +-                   -+
{                     |                     |
{                     |  u^3  u^2   u   1   |
{                     |                     |
{                     +-                   -+
{
{          Mh     is the Hermite matrix which is:
{                     +-                   -+
{                     |   2   -2    1    1  |
{                     |  -3    3   -2   -1  |
{                     |   0    0    1    0  |
{                     |   1    0    0    0  |
{                     +-                   -+
{
{          Qh     is the geometry matrix as calculated above
{
{          (Mh)t  is the transpose of the Hermite matrix
{
{          (V)t   is the transpose of the matrix:
{                     +-                   -+
{                     |                     |
{                     |  v^3  v^2   v   1   |
{                     |                     |
{                     +-                   -+
{
{  and the definition of the Bezier surface is:
{
{    P(u,v) := U * Mb * Qb * (Mb)t * (V)t     0 <= u,v <= 1
{
{  Where the definitions are as above and additionally:
{
{          Mb     is the Bezier matrix which is:
{                     +-                   -+
{                     |   1    3   -3    1  |
{                     |   3   -6    3    0  |
{                     |  -3    3    0    0  |
{                     |   1    0    0    0  |
{                     +-                   -+
{
{          (Mb)t  is the transpose of the Bezier matrix
{                 ( it happens that Mb = (Mb)t )
{
{          Qb     is the Bezier control point matrix
{
{  Then:
{
{    U * Mb * Qb * (Mb)t * (V)t = U * Mh * Qh * (Mh)t * (V)t
{
{    Mb * Qb * (Mb)t  =  Mh * Qh * (Mh)t
{
{    Qb * (Mb)t = Mb-^1 * Mh * Qh * (Mh)t
{
{    Qb = Mb-^1  * Mh * Qh * (Mh)t * ((Mb)t)-^1
{                                                                   }
{                                                                   }
{  Note that:                                                       }
{                                                                   }
{          Mb-¹ is the inverse of Mb which is:                      }
{                    +-                   -+                        }
{                    |  0    0    0    3   |                        }
{               %1   |   0    0    1    3   |
}
{               3   |   0    1    2    3   |                        }
{                   |   3    3    3   -3   |                        }
{                   +-                   -+                         }
{                                                                   }
{          let A = Mb-¹ * Mh  which equals:                         }
{                   +-                   -+                         }
{                   |   3    0    0    0   |                        }
{               %1   |   3    0    1    0   |
}
{               3   |   0    3    0   -1   |                        }
{                   |  -6    3    0    0   |                        }
{                   +-                   -+                         }
{                                                                   }
{          and ((Mb)t)-¹ is the inverse of (Mb)t which is:
{                   +-                   -+
{                   |   0    0    0    3   |
{               %1   |   0    0    1    3   |
}
{               3   |   0    1    2    3   |
{                   |   3    3    3   -3   |
{                   +-                   -+
{
{          let B = (Mh)t * ((Mb)t)-¹ which equals:
{                   +-                   -+
{                   |   3    3    0   -6   |
{               %1   |   0    0    3    3   |
}
{               3   |   0    1    0    0   |
{                   |   0    0   -1    0   |
{                   +-                   -+
{                     (notice B = (A)t)
{
{
{   Therefore  Qb = A * Qh * (A)t
{
{


point_matrix_x := A * temp_matrix_x * (A)t;
point_matrix_y := A * temp_matrix_y * (A)t;
point_matrix_z := A * temp_matrix_z * (A)t;

for i1 := 1 to 4 do
  for i2 := 1 to 4 do begin
    CARTESIAN_THREE_COORDINATE.X := point_matrix_x[ i1, i2 ];
    CARTESIAN_THREE_COORDINATE.Y := point_matrix_y[ i1, i2 ];
    CARTESIAN_THREE_COORDINATE.Z := point_matrix_z[ i1, i2 ];
    create CARTESIAN_THREE_COORDINATE[ i1, i2 ];
    BEZIER_SURFACE.CONTROL_POINTS[i1,i2] :=
                   CARTESIAN_THREE_COORDINATE[ i1, i2 ]
  end;

BEZIER_SURFACE.U_RATIONAL := FALSE;
BEZIER_SURFACE.V_RATIONAL := FALSE;

BEZIER_SURFACE.U_UNIFORM := TRUE;
BEZIER_SURFACE.V_UNIFORM := TRUE;

BEZIER_SURFACE.U_DEGREE := 3;
BEZIER_SURFACE.V_DEGREE := 3;

BEZIER_SURFACE.U_UPPER := 3;
BEZIER_SURFACE.V_UPPER := 3;

BEZIER_SURFACE.WEIGHTS := 0;
BEZIER_SURFACE.FORM_NUMBER := o;

create BEZIER_SURFACE;
```

7.12.2 From a Bezier surface to a Hermitian surface

```
{  The PDES BEZIER SURFACE entity will be tranlated into a
{  Hermitian surface or patch defined by 4 Hermite cubics, of
{  degree 3, and a Hermitian surface or patch.
{    1        CUBIC PATCH
{    4        CUBIC CURVEs

if ( BEZIER_SURFACE.U_RATIONAL = False )and
   ( BEZIER_SURFACE.U_RATIONAL = False )and
   ( BEZIER_SURFACE.U_UNIFORM  = True  )and
   ( BEZIER_SURFACE.U_UNIFORM  = True  )and
   ( BEZIER_SURFACE.U_DEGREE   = 3     )and
   ( BEZIER_SURFACE.V_DEGREE   = 3     )and
   ( BEZIER_SURFACE.U_UPPER    = 3     )and
   ( BEZIER_SURFACE.V_UPPER    = 3     )
  then begin

    for i1 := 0 to 3 do begin
      for i2 := 0 to 3 do begin

        B_matrix_x[ i1, i2 ] :=
                BEZIER_SURFACE.CONTROL_POINTS[ i1, i2 ]->
                CARTESIAN_THREE_COORDINATE.X;
        B_matrix_y[ i1, i2 ] :=
                BEZIER_SURFACE.CONTROL_POINTS[ i1, i2 ]->
                CARTESIAN_THREE_COORDINATE.Y;
        B_matrix_z[ i1, i2 ] :=
                BEZIER_SURFACE.CONTROL_POINTS[ i1, i2 ]->
                CARTESIAN_THREE_COORDINATE.Z
```

```
     end
   end;

{                                                                    }
{  Since the definition of the Bezier cubic surface is:              }
{    P(u,v) := U * Mb * Qb * (Mb)t * (V)t     0 <= u,v <= 1          }
{                                                                    }
{  Where   U      is the matrix:                                     }
{                      +-                 -+                         }
{                      |                   |                         }
{                      |  u^3  u^2   u   1 |                         }
{                      |                   |                         }
{                      +-                 -+                         }
{                                                                    }
{          Mb     is the Bezier matrix which is:                     }
{                      +-                 -+                         }
{                      |   1    3   -3   1 |                         }
{                      |   3   -6    3   0 |                         }
{                      |  -3    3    0   0 |                         }
{                      |   1    0    0   0 |                         }
{                      +-                 -+                         }
{                                                                    }
{          Qb     is the geometry matrix as calculated above         }
{                                                                    }
{          (Mb)t  is the transpose of the Bezier matrix              }
{                 ( it happens that Mb = (Mb)t )                     }
{                                                                    }
{          (V)t   is the transpose of the matrix:                    }
{                      +-                 -+                         }
{                      |                   |                         }
{                      |  v^3  v^2   v   1 |                         }
{                      |                   |                         }
{                      +-                 -+                         }
{                                                                    }
{  and the definition of the Hermite cubic surface (CADD             }
{  PATCH) is:                                                        }
{                                                                    }
{    P(u,v) := U * Mh * Qh * (Mh)t * (V)t     0 <= u,v <= 1          }
{                                                                    }
{  Where the definitions are as above and additionally:              }
{                                                                    }
{          Mh     is the Hermite matrix which is:                    }
{                      +-                 -+                         }
{                      |   2   -2    1   1 |                         }
{                      |  -3    3   -2  -1 |                         }
{                      |   0    0    1   0 |                         }
{                      |   1    0    0   0 |                         }
{                      +-                 -+                         }
{          (Mh)t  is the transpose of the Hermite matrix             }
{                                                                    }
{          Qh     is the Hermite geometry matrix:                    }
{   +-                                                           -+  }
{   |                                :                            |  }
{   |   P(0,0)     P(0,1)            :    %__dx__%      %__dx__%     |
}
```

```
{     |                                 :   dv(0,0)      dv(0,1)    |
{     |                                 :                           |
{     |   P(1,0)      P(1,1)            :   %__dx__%      %__dx__%
}
{     |                                 :   dv(1,0)      dv(1,1)    |
{     | ................................:........................   |
{     |                                 :                           |
{     |   %__dx__%        %__dx__%        :   %__d²x__%      %__d²x__%
|   }
{     |   du(0,0)       du(0,1)         :  dudv(0,0)    dudv(0,1)   |
{     |                                 :                           |
{     |   %__dx__%        %__dx__%        :   %__d²x__%      %__d²x__%
|   }
{     |   du(1,0)       du(1,1)         :  dudv(1,0)    dudv(1,1)   |
{     |                                 :                           |
{     +-                                                           -+
{
{   for the X coordinate.  And is similar for the Y and Z
{   coordinates.
{
{   Then:
{
{     U * Mh * Qh * (Mh)t * (V)t = U * Mb * Qb * (Mb)t * (V)t
{
{     Mh * Qh * (Mh)t  =  Mb * Qb * (Mb)t
{
{     Qh * (Mh)t = Mh-¹ * Mb * Qb * (Mb)t
{
{     Qh = Mh-¹ * Mb * Qb * (Mb)t * ((Mh)t)-¹
{
{
{   Note that:
{
{           Mh-¹ is the inverse of Mh which is:
{                          +-                -+
{                          |   0   0   0   1  |
{                          |   1   1   1   1  |
{                          |   0   0   1   0  |
{                          |   3   2   1   0  |
{                          +-                -+
{
{           let A = Mh-¹ * Mb  which equals:
{                          +-                -+
{                          |   1   0   0   0  |
{                          |   2   0   0   1  |
{                          |  -3   3   0   0  |
{                          |   6   0  -3   3  |
{                          +-                -+
{
{           and ((Mh)t)-¹ is the inverse of (Mh)t which is:
{                          +-                -+
{                          |   0   1   0   3  |
{                          |   0   1   0   2  |
{                          |   0   1   1   1  |
{                          |   1   1   0   0  |
{                    +-                  -+                         }
{                                                                   }
{        let B = (Mb)t * ((Mh)t)-¹ which equals:                    }
{                    +-                  -+                         }
{                    |   1   2  -3   6   |                          }
{                    |   0   0   3   0   |                          }
{                    |   0   0   0  -3   |                          }
{                    |   0   1   0   3   |                          }
{                    +-                  -+                         }
{                     (notice B = (A)t)                             }
{                                                                   }
{                                                                   }
{   Therefore  Qh = A * Qb * (A)t                                   }
{                                                                   }

    H_matrix_x := A * B_matrix_x * (A)t;
    H_matrix_y := A * B_matrix_y * (A)t;
    H_matrix_z := A * B_matrix_z * (A)t;

{                                                                   }
{   Construct the U=0 Cubic curve                                   }
{                                                                   }
    CUBIC[1].P0.X := H_matrix_x[0,0];
    CUBIC[1].P0.Y := H_matrix_y[0,0];
    CUBIC[1].P0.Z := H_matrix_z[0,0];

    CUBIC[1].V0.I := H_matrix_x[0,2];
    CUBIC[1].V0.J := H_matrix_y[0,2];
    CUBIC[1].V0.K := H_matrix_z[0,2];

    CUBIC[1].V1.I := H_matrix_x[0,3];
    CUBIC[1].V1.J := H_matrix_y[0,3];
    CUBIC[1].V1.K := H_matrix_z[0,3];

    CUBIC[1].P1.X := H_matrix_x[0,1];
    CUBIC[1].P1.Y := H_matrix_y[0,1];
    CUBIC[1].P1.Z := H_matrix_z[0,1];

    create CUBIC[1];
    PATCH.U0CRV := CUBIC[1];

{                                                                   }
{   Construct the U=1 Cubic curve                                   }
{                                                                   }

    CUBIC[2].P0.X := H_matrix_x[1,0];
    CUBIC[2].P0.Y := H_matrix_y[1,0];
    CUBIC[2].P0.Z := H_matrix_z[1,0];

    CUBIC[2].V0.I := H_matrix_x[1,2];
    CUBIC[2].V0.J := H_matrix_y[1,2];
    CUBIC[2].V0.K := H_matrix_z[1,2];

    CUBIC[2].V1.I := H_matrix_x[1,3];
    CUBIC[2].V1.J := H_matrix_y[1,3];
    CUBIC[2].V1.K := H_matrix_z[1,3];

    CUBIC[2].P1.X := H_matrix_x[1,1];
    CUBIC[2].P1.Y := H_matrix_y[1,1];
    CUBIC[2].P1.Z := H_matrix_z[1,1];

    create CUBIC[2];
    PATCH.U1CRV := CUBIC[2];

{
{   Construct the V=0 Cubic curve
{

    CUBIC[3].P0.X := H_matrix_x[0,0];
    CUBIC[3].P0.Y := H_matrix_y[0,0];
    CUBIC[3].P0.Z := H_matrix_z[0,0];

    CUBIC[3].V0.I := H_matrix_x[2,0];
    CUBIC[3].V0.J := H_matrix_y[2,0];
    CUBIC[3].V0.K := H_matrix_z[2,0];

    CUBIC[3].V1.I := H_matrix_x[3,0];
    CUBIC[3].V1.J := H_matrix_y[3,0];
    CUBIC[3].V1.K := H_matrix_z[3,0];

    CUBIC[3].P1.X := H_matrix_x[1,0];
    CUBIC[3].P1.Y := H_matrix_y[1,0];
    CUBIC[3].P1.Z := H_matrix_z[1,0];

    create CUBIC[3];
    PATCH.V0CRV := CUBIC[3];

{                                                                        }
{   Construct the V=1 Cubic curve                                        }
{                                                                        }

    CUBIC[4].P0.X := H_matrix_x[0,1];
    CUBIC[4].P0.Y := H_matrix_y[0,1];
    CUBIC[4].P0.Z := H_matrix_z[0,1];

    CUBIC[4].V0.I := H_matrix_x[2,1];
    CUBIC[4].V0.J := H_matrix_y[2,1];
    CUBIC[4].V0.K := H_matrix_z[2,1];

    CUBIC[4].V1.I := H_matrix_x[3,1];
    CUBIC[4].V1.J := H_matrix_y[3,1];
    CUBIC[4].V1.K := H_matrix_z[3,1];

    CUBIC[4].P1.X := H_matrix_x[1,1];
    CUBIC[4].P1.Y := H_matrix_y[1,1];
    CUBIC[4].P1.Z := H_matrix_z[1,1];

    create CUBIC[4];
    PATCH.V1CRV := CUBIC[4];
```

```
{                                                             }
{   Construct the twist vectors for the patch                 }
{                                                             }

    PATCH.NU0V0.I  := H_matrix_x[3,3];
    PATCH.NU0V0.J  := H_matrix_y[3,3];
    PATCH.NU0V0.K  := H_matrix_z[3,3];

    PATCH.NU0V1.I  := H_matrix_x[3,4];
    PATCH.NU0V1.J  := H_matrix_y[3,4];
    PATCH.NU0V1.K  := H_matrix_z[3,4];

    PATCH.NU1V0.I  := H_matrix_x[4,3];
    PATCH.NU1V0.J  := H_matrix_y[4,3];
    PATCH.NU1V0.K  := H_matrix_z[4,3];

    PATCH.NU1V1.I  := H_matrix_x[4,4];
    PATCH.NU1V1.J  := H_matrix_y[4,4];
    PATCH.NU1V1.K  := H_matrix_z[4,4];

    create PATCH

  end
  else
    (* unable at this time to convert *)
  end
```

7.13 Conversion Issues

1. What is the best method for converting multisegment BEZIER Curves?

2. Although not strictly an issue, the functions: USERS, SQRT, and POINTONCURVE and the operators: "**", "in", and "->" need to be defined a the top of the document.

8.0 COMPATIBILITY WITH OTHER STANDARDS

This chapter will describe the relationship between this standard and other related standards.

```
-------   +-------------------------------------------------------------+
   A      |                                                             |
   |      |           STEP reference model (Document 3.)                |
   |      |                                                             |
   |      +-------------------------------------------------------------+
   |
   |      +-------------------------------------------------------------+
   |      |       Sequentialised representation according to           |
  STEP    |         document 4.2.1, 4.2.2                               |
   |      |                                                             |
   |      +-------------------------------------------------------------+
   |
   |      +-------------------------------------------------------------+
   |      |                    encoding formats                         |
   |      |-------------------------------------------------------------|
   |      |    clear text     | character encoding|  binary encoding    |
   |      |     encoding      |                   |                     |
   |      |                   |   to be defined   |   to be defined     |
   V      | see appendix ??   |                   |                     |
-------   +-------------------------------------------------------------+

-------   +-------------------------------------------------------------+
   A      | international file exchange|   private file exchange        |
   |      | based on existing standards|                                |
   |      | for the application layer  |                                |
application|                           |--------------------------------|
  layer   |- FTAM ISO 8571             |                                |
   |      |- CCITT X.400               |   ( STEPFILE )                 |
   |      |- MAP/TOP                   |                                |
   |      |- ODA/ODIF                  |                                |
   |      +-------------------------------------------------------------+
   |      |        transparent         |      transparent for           |
   |      |                            |    basic     |     all         |
   |      |                            |  character   |    8-bit        |
   |      |                            |set only, for| combinations,    |
   |      |                            | clear text  |   for all        |
   V      |                            |encoding only|  encodings       |
-------   +-------------------------------------------------------------+
   A      |                            |             |                  |
   |      |           see              |    see      |     see          |
   |      |       corresponding        |   D.2.1     |     D.2          |
  lower   |         standards          |             |                  |
  layers  |                            |             |                  |
   |      |                            |             |                  |
   V      |                            |             |                  |
-------   +-------------------------------------------------------------+
```

D.2 ENCODING SCHEMES

D.2.1 CLEAR TEXT ENCODING

Two representations schemes are provided: 7-bit representations and 8-bit representations. Both conform to ISO 646 and ISO 6937/1. In both encoding schemes only characters with decimal code values 32 through 126 (printable characters) corresponding to the basic character set are utilized.

D.2.1.1 7-BIT REPRESENTATION OF THE BASIC CHARACTER SET

b7-------->				0	0	0	0	1	1	1	1
b6-------->				0	0	1	1	0	0	1	1
b5-------->				0	1	0	1	0	1	0	1
b4	b3	b2	b1								
0	0	0	0			32	48	64	80	96	112
0	0	0	1			33	49	65	81	97	113
0	0	1	0			34	50	66	82	98	114
0	0	1	1			35	51	67	83	99	115
0	1	0	0			36	52	68	84	100	116
0	1	0	1			37	53	69	85	101	117
0	1	1	0			38	54	70	86	102	118
0	1	1	1			39	55	71	87	103	119
1	0	0	0			40	56	72	88	104	120
1	0	0	1			41	57	73	89	105	121
1	0	1	0			42	58	74	90	106	122
1	0	1	1			43	59	75	91	107	123
1	1	0	0			44	60	76	92	108	124
1	1	0	1			45	61	77	93	109	125
1	1	1	0			46	62	78	94	110	126
1	1	1	1			47	63	79	95	111	

The table enties indicate the decimal codee character values of the basic character set (see B.1).
Any entries in this table which are left blank do not conform to this standard.

D.2.1.2 8-BIT REPRESENTATION OF THE BASIC CHARACTER SET

The 8-bit representations are identical to the 7-bit representations with bit b8 set to 0. Bit representations with bit b8 set to 1 do not conform to this standard.

D.2.2 CHARACTER ENCODING

Similar to the encoding schemes of the Computer Graphics Metafile (ISO 8632) a more compact character encoding scheme will be developed.

D.2.3 BINARY ENCODING

Similar to the encoding schemes of the Computer Graphics Metafile (ISO 8632) a more compact binary encoding scheme will be developed.

D.3 STEP FILE EXCHANGE BASED ON INTERNATIONAL APPLICATION LAYER STANDARDS

In an environment which supports standard file transfer (FTAM) or standard interpersonal message exchange services (CCITT-X.400) these services are utilized irrespectively for all STEP encoding schemes. The network layers below the application layer are of no concern to the STEP file exchange as full transparency of the transfer is provided.

D.3.1 STEP FILE EXCHANGE IN AN FTAM ENVIRONMENT

A sequence of STEP files (STEPFILE) is considered as a single unstructured file to be transferred with the standard FTAM service.

D.3.2 STEP FILE EXCHANGE IN AN CCITT-X.400 ENVIRONMENT

A sequence of STEP files (STEPFILE) is considered as a single message file to be transferred with the standard CCITT-X.400 service.

D.3.3 STEP FILE EXCHANGE IN A MAP/TOP ENVIRONMENT

In a MAP/TOP environment FTAM and/or CCITT-X.400 services may be utilized for transfer of sequences of STEP files.

D.3.4 STEP FILE EXCHANGE IN AN ODA/ODIF ENVIRONMENT

In an ODA/ODIF environment a single STEP file is treated as a single office document.

D.4. PRIVATE STEP FILE EXCHANGE (NOT NECESSARILY TRANSPARENT)

Private STEP file exchange requires specification of the lower levels of the computer interconnection architecture. In all cases, on the application layer a sequence of STEP files (STEPFILE) is exchanged.

application layer	(STEPFILE)	
presentation layer	logical record length = 80 characters see 1) below	
session layer	empty	
transport layer	block oriented transport	line oriented transport
	block length: 800 characters last block padded to 800 see 2) below	80 character lines separated by CR,LF terminated by (CTRL Z) see 3) below
network layer	empty	empty
link layer	empty	empty
physical layer	see 4) below	see 5) below

Remarks:

1) The first character in the file must be the "S of the STEP file reserved word "STEP;".
The last record must be padded (if necessary) with SPACE (DCC = 32).

2) The first character in the file must be the "S of the STEP file reserved word "STEP;".
The last block must be padded (if necessary) with SPACE (DCC = 32).

3) The sequence CR,LF is represented by bit combinations 0/13 0/10 (DCC = 13 followed by DCC = 10)
The last line (padded to 80 characters if necessary) is not followed by CR,LF, but ratherby (CTRL Z) represented by the bit combination 1/10 (DCC = 26).

4) Possible Storage Media for block oriented transport
   a) Tape
      * 1/2 INCH TAPE
      * STANDARD
      * UNLABELLED
      * 9 TRACK
      * 1600 BPI
      * PHYSICAL BLOCKSIZE 800
        blocks separated by inter-record gaps,
        last block followed by tape mark
   b) Other physical storage media (disk, e.g.) compatible with this transport format
   c) private computer network compatible with this transport format

5) Possible Storage Media for line oriented transport
   a) 3 1/2" diskette
   b) 5 1/4" diskette
   c) Other physical storage media (disk, e.g.) compatible with this transport format
   d) private computer network compatible with this transport format

D.5. STEP FILE EXCHANGE IN AN ISO 2022 CODING ENVIRONMENT

For exchange with services using the code extension techniques of ISO 2022, the STEP file coding environment shall be invoked from the ISO 2022 environment by the following escape sequence:

ESC 2/5 S

where ESC is the bit combination 1/11 (DCC = 25)
2/5 is the bit combination with DCC = 37
S refers to a bit copmbination (with corresponding DCC) that will be assigned by the ISP Registration Authority for CODE Extensions and will be substituted in this Standard prior to publication.

After the end of one or more STEP files (after "ENDSTEP;" followed possibly by padding with SPACE, DCC=32) the following escape sequence may be used to return to the ISO 2022 coding environment:

ESC 2/5 4/0

This not only returns to the ISO 2022 coding environment, but also restores the designation and invocation of poded character sets to the state that existed prior to entering the STEP coding environment with the ESC 2/5 S sequence. (The terms "designation" and "invocation" are defined in ISO 2022).

ANNEX D (Informative)

SECTION 2

LIBRARY OF STEP TEST PARTS

ANNEX D (Informative)

SECTION 3

SOFTWARE TOOLS

ANNEX D (Informative)

SECTION 4

SPECIAL TREATISES

Title: SG6 Issues Log

Owner: P.R. Wilson

Date: 24 October 1988

Corresponding ISO Document Number: N290

N289

ISO TC184/SC4/WG1 **Document** October 24, 1988

N290 (Working)

EXTERNAL REPRESENTATION OF PRODUCT DEFINITION DATA

DOCUMENT NUMBER 4 **Version** Tokyo

TITLE: SG6 Issue Log

ABSTRACT: This document is the Issue Log for the Testing Draft.

KEY WORDS: Integrated Model, EXPRESS, Issues

DATE: October 24, 1988

OWNER: SG6,
Editor P. R. Wilson
General Electric CR&D
Building K1, Room KW-D238
River Road
Schenectady, NY 12309
U.S.A

TELEPHONE: (518) 387-5241

TELEX :

ISO REPRESENTATIVE: Peter Wilson

STATUS: Working

SG6 Issue Log

Peter R. Wilson

October 24, 1988

This document contains a description of the issues concerning the Integrated Information Model (IPIM) as rendered in the Initial Testing Draft (ITD).

LOG of OPEN and RESOLVED ISSUES

Issue	Title	Date Described	Date Resolved
SG6 – 1	Mapping syntax	04/22/88	07/14/88
SG6 – 2	Superfluous TYPE definitions	04/22/88	03/31/88
SG6 – 3	Use of OPERATION	04/22/88	03/31/88
SG6 – 4	Use of GENERIC	04/22/88	07/14/88
SG6 – 5	SUPERTYPE only for attribute inheritance	04/22/88	07/14/88
SG6 – 6	Extensions to EXPRESS	07/05/88	07/14/88
SG6 – 7	Non-conformance to ISO standards	07/05/88	07/14/88
SG6 – 8	Location of TYPE definitions	07/05/88	03/31/88
SG6 – 9	Cross-reference listing	07/05/88	10/17/88
SG6 – 10	Uncomputable FUNCTIONs	08/15/88	10/17/88

ISSUE SG6 – 1 Discussion of Mapping Syntax is unclear.

INITIATION DATE: 22 April 1988

INITIATOR: Chia-Hui Shih

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: Several points were raised about the description of the Mapping Syntax in the Washington(?) edition of the Test Draft.

DISCUSSION:

Since the Washington Edition the EXPRESS language has been updated to include a MAP syntax.

OPTION PROPOSED: This Section of the ITD will be deleted in the West Palm Beach edition.

EXPLANATION: Mapping is now part of EXPRESS.

DECISION DATE: 14 July 1988

IMPLEMENTATION RESPONSIBILITY: Wilson and Kennicott.

ISSUE SG6 – 2 Superfluous TYPE definitions.

INITIATION DATE: 22 April 1988

INITIATOR: Chia-Hui Shih

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: A desire to have TYPE definitions such as

```
CURVE_OR_SURFACE = SELECT(CURVE, SURFACE)
```

defined "in-line" as part of an ATTRIBUTE statement.

DISCUSSION:

This topic has been the subject of discussions between the Modelers and the developers of EXPRESS. In-line SELECTs etc. have never been part of EXPRESS and will not be.

OPTION PROPOSED: None.

EXPLANATION: Issue resolved by EXPRESS Committee.

DECISION DATE: 31 March 1988

IMPLEMENTATION RESPONSIBILITY: None.

ISSUE SG6 – 3 Use of OPERATION.

INITIATION DATE: 22 April 1988

INITIATOR: Chia-Hui Shih

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: Comments about the use of defined OPERATIONS.

DISCUSSION:

The OPERATION was removed from EXPRESS in N210. Consequently, OPERATION definitions did not appear in the Denver ITD.

OPTION PROPOSED: None.

EXPLANATION: OPERATION is no longer.

DECISION DATE: 31 March 1988

IMPLEMENTATION RESPONSIBILITY: None.

ISSUE SG6 – 4 Use of GENERIC.

INITIATION DATE: 22 April 1988

INITIATOR: Chia-Hui Shih

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: Many generic arguments are used in functions where the functions only make sense for some specific types of argument.

DISCUSSION:

Some functions are generic in nature; others only make sense for specific types of arguments. However, as new Topical Models are still being added to the ITD, it is not yet clear exactly what the appropriate sets of argument types should be.

Developing the IPIM is like developing a computer program. The use of generic types as function arguments is similar to putting null procedures in a program as stubs for later definition.

OPTION PROPOSED: Specify all arguments correctly when the decision is made as to what will be in STEP version 1.

EXPLANATION:

DECISION DATE: 14 July 1988

IMPLEMENTATION RESPONSIBILITY: SG6 and "owners" of FUNCTIONs.

ISSUE SG6 - 5 SUPERTYPING should only be used to provide attribute inheritance.

INITIATION DATE: 22 April 1988

INITIATOR: Chia-Hui Shih

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: There is a perception that there was an agreement in St Louis that SUPERTYPES (as entities) can be only used to provide attribute inheritance. In the IPIM they are extensively used in place of SELECT.

DISCUSSION:

A SUPERTYPE is used to group together classes of things that to the information modeler have something in common — ideas and/or attributes. The commonality is independent of any context (usage) of the grouping.

A SELECT is used as a context dependent selection among things that may or may not have context-independent commonality.

It often happens that a pre-existing SUPERTYPE provides a "ready made" selection and then there is no point in defining a new SELECT type.

It should be noted that the IPIM is independent of any particular type of implementation (e.g file transfer) and should support all types of implementation (e.g levels 1 through 4). Constraints imposed by a particular type of implementation should not be imposed on the IPIM, but have to be resolved in mapping from the IPIM to the particular implementation.

OPTION PROPOSED: Do nothing.

EXPLANATION: SUPERTYPE and SELECT have been used appropriately within the IPIM.

DECISION DATE: 14 July 1988

IMPLEMENTATION RESPONSIBILITY: None.

ISSUE SG6 – 6 Extension to the EXPRESS language should be removed.

INITIATION DATE: 5 July 1988

INITIATOR: The Netherlands

STATUS: Resolved

RELATED ISSUES: SG6 – 1

DESCRIPTION: Extensions to the EXPRESS language should be removed from document WG1/N224 and put into the EXPRESS language document.

DISCUSSION:

Agreed. This Section is being deleted from the ITD.

OPTION PROPOSED: Delete the EXPRESS section from the ITD.

EXPLANATION:

DECISION DATE: 14 July 1988

IMPLEMENTATION RESPONSIBILITY: Wilson and Kennicott.

ISSUE SG6 – 7 Non-conformance of the document to ISO standards.

INITIATION DATE: 5 July 1988

INITIATOR: Germany

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: The ITD as balloted on does not conform to ISO documentation standards.

DISCUSSION:

Agreed.

The ITD reflects the current work status within PDES/STEP and technical issues have taken precedence over documentation standards. The next version of the ITD to be voted on will conform to ISO documentation standards.

OPTION PROPOSED: The Tokyo edition of the ITD will conform to ISO documentation standards.

EXPLANATION:

DECISION DATE: 14 July 1988

IMPLEMENTATION RESPONSIBILITY: SG6

ISSUE SG6 – 8 Location of TYPE and FUNCTION definitions.

INITIATION DATE: 5 July 1988

INITIATOR: Germany

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: TYPEs and FUNCTIONs should be defined within the appropriate model sections.

DISCUSSION:

Agreed.

One of the changes from the Washington to the Denver edition of the ITD was the relocation of these definitions, as proposed.

OPTION PROPOSED: Relocation in hand.

EXPLANATION:

DECISION DATE: 31 March 1988

IMPLEMENTATION RESPONSIBILITY: Wilson and Kennicott

ISSUE SG6 – 9 Provide a full Cross-reference listing.

INITIATION DATE: 5 July 1988

INITIATOR: Germany

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: The classification should be extended to a full cross reference listing.

DISCUSSION:

There is no problem in doing this now that EXPRESS report writers are available. The question is one of space — a full cross reference listing will require 100–200 pages and the ITD already has 700–800 pages. Is it worth it?

OPTION PROPOSED: Provide a cross reference listing after the Conceptual Schema has been put under configuration control, which is expected to occur for the Tokyo edition (after the Tokyo meeting).

EXPLANATION: There seems little point in increasing the documentation bulk until a reasonably stable state has been reached.

DECISION DATE: 17 October 1988

IMPLEMENTATION RESPONSIBILITY: SG6

ISSUE SG6 – 10 Uncomputable FUNCTIONS

INITIATION DATE: 15 August 1988

INITIATOR: Various

STATUS: Resolved

RELATED ISSUES:

DESCRIPTION: In many instances throughout the IPIM FUNCTIONs appear within WHERE clauses to define constraints. In some cases no code is defined for the body of the FUNCTION. In these cases the FUNCTION should either be deleted or code provided.

DISCUSSION:

The constraints on an entity are just as vital a part of its definition as the attributes which compose the entity. These must both be defined in a computer processible manner. There are, however, some constraints that are not necessarily computable or, if they are computable, would require enormous effort to define the algorithm.

Within the IPIM there are four classes of FUNCTION:

1. Computable and code provided. For example, the function **occurs** which, given a "name" and a list of "names" returns the number of occurrences of the given name in the list.
2. Readily computable and no code provided. For example, the function **cross product** which takes two input vectors and returns their cross product.
3. Computable but algorithmically a major task. For example, the function **arc length extent** which calculates the arc length of a **curve**. Developing the code body for this function would take multiple man-months.
4. Uncomputable. An example of this type of function is **manifold** which should determine if its input argument is a manifold or not.

Most functions have been defined so that they return "simple" results (e.g a LOGICAL or REAL value) and are used within a correspondingly simple context (e.g does the REAL value fall within a given range?). As an example, the typical usage of **arc length extent** is to say that particular forms of a STEP curve must not have zero length. If this is a required constraint, then it must be included within the EXPRESS definition of the **curve**.

OPTIONS

1. Delete all uncoded FUNCTIONS.
2. Replace all uncoded constraints by comments.
3. Code all computable FUNCTIONS and replace uncomputable constraints by comments.
4. Code all easily computable FUNCTIONS and leave others "as is".

OPTION PROPOSED: Provide text definitions of the intended result(s) for each function; do not provide code for "standard functions" (e.g scalar product of two vectors); provide code for the remainder of easily computable functions; do not provide code for complex or uncomputable functions. This issue should also be addressed in the *Style Guide*.

EXPLANATION: Computer processible constraints, computable or not, are an essential part of the Information Model. Comments are not computer processible.

DECISION DATE: 17 October 1988

IMPLEMENTATION RESPONSIBILITY: SG6 and owners of FUNCTIONS and constraints.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 88-4004	2. Performing Organ. Report No.	3. Publication Date DECEMBER 1988

4. TITLE AND SUBTITLE

Product Data Exchange Specification
First Working Draft

5. AUTHOR(S)

Bradford Smith, Gaylen Rinaudot

6. PERFORMING ORGANIZATION (*If joint or other than NBS, see instructions*)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Interim Report

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (*Street, City, State, ZIP*)

NIST
Bldg. 220 Room A-150
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (*A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here*)

This document contains a neutral format for the representation and communication of product data. Known as the Product Data Exchange Specification (PDES), the document had been developed by the IGES/PDES Organization with active cooperation from the Working Group 1 of ISO/TC184/SC4. It represents the first working draft of PDES for presentation to the international community.

12. KEY WORDS (*Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons*)

Product Data, PDES, Computer-integrated manufacturing (CIM)

13. AVAILABILITY

☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

1991

15. Price

$126.95

AMERICAN BINDERY
2
5-94